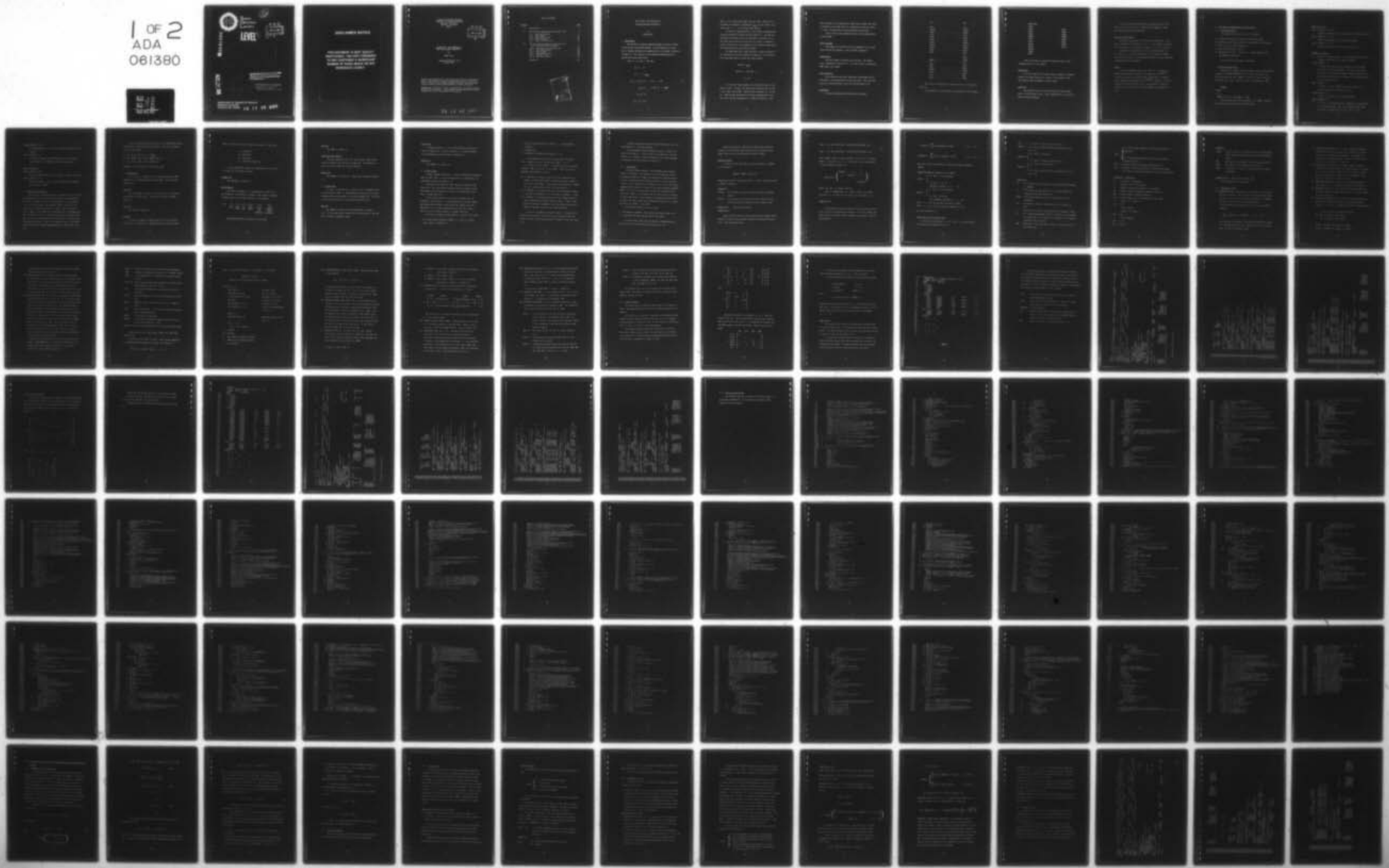
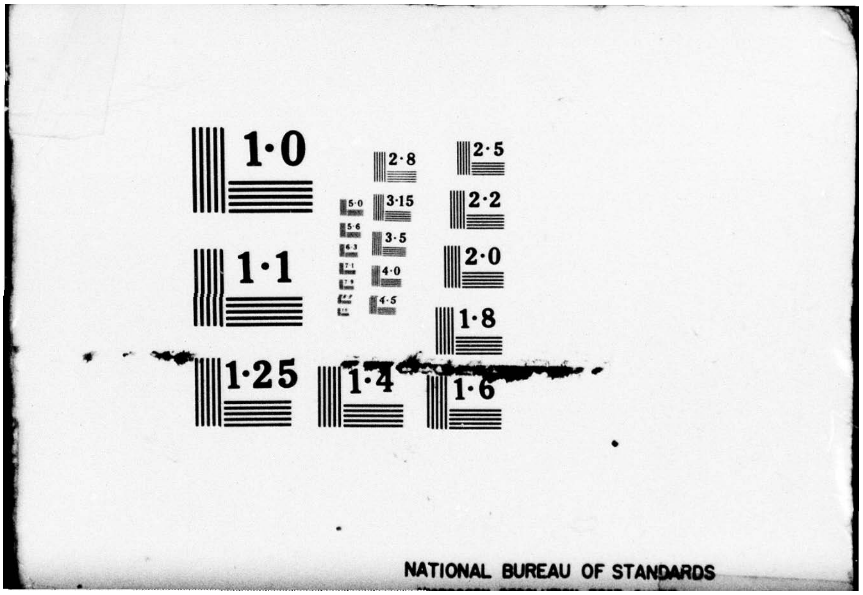
AD-A061 380 STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB F/G 5/3
BCA AND HRA: TWO PROGRAMS FOR COMPUTING ECONOMIC EQUILIBRIA.(U)
AUG 78 T ELKEN N00014-75-C-0267
UNCLASSIFIED SOL-78-17 NL

1 OF 2
ADA
061380




1·0
2·8
2·5
5·0
3·15
2·2
5·6
6·3
3·5
7·1
1·1
4·0
2·0
4·5
1·8
1·25
1·4
1·6
NATIONAL BUREAU OF STANDARDS

12

B.S.

Systems
Optimization
Laboratory

LEVEL II

Department of Operations Research
Stanford University
Stanford, CA 94305

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
Stanford University
Stanford, California
94305

BCA AND HRA: TWO PROGRAMS FOR COMPUTING ECONOMIC EQUILIBRIA

by

Thomas Elken

TECHNICAL REPORT SOL 78-17
August 1978

Research and reproduction of this report were partially supported by the U.S. Department of Energy Contract EY-76-S-03-0326 PA #18; The Office of Naval Research Contract N00014-75-C-0267; and the National Science Foundation Grants MCS76-20019 A01 and MCS76-81259 A01.

TABLE OF CONTENTS

Section | Page

I Introduction 1

II The Bilinear Complementarity Algorithm (BCA) 7

II.1. Input Requirements 7
II.2. Main Program 15
II.3. Subroutines of BCA 21
II.4. Sample Problems 29
II.5. Source Listing for BCA 44

III The HRA (Homotopy Retraction Algorithm) Code for Solving Equilibrium Problems 79

III.1. Revisions to the Original Code 79
III.2. Input Requirements 82
III.3. Main Program 83
III.4. Subroutines of HRA 85
III.5. Sample Problems 89
III.6. HRA Source Listing 96

References 128

BCA and HRA: Two Programs for Computing Economic Equilibria

by

Thomas Elken

I. Introduction

BCA and HRA are computer programs designed to solve a version of the economic equilibrium problem. For an introduction to this type of problem and models for dealing with it, the reader is directed to Elken [2]. Very briefly, we will describe mathematically the problem which these codes solve.

Find x, t, λ, and ζ such that

$$
\begin{aligned}
&Dx + t = b \\
&\lambda D - \zeta = e_{NCOL} \\
&\langle C_i, \lambda \rangle - \lambda_i t_i = 0 , \quad i = 1, \ldots, IH \\
&\lambda_i t_i = 0 , \quad i = IH + 1, \ldots, NROW \\
&\langle x, \zeta \rangle = 0 \\
&x, t, \lambda, \zeta \geq 0 ,
\end{aligned}
\tag{1}
$$

where D is a matrix with NROW rows and NCOL columns, IH is the number of consumers or households, e_{NCOL} is the NCOLth unit vector, and $\langle \cdot, \cdot \rangle$ is the usual inner product.

The BCA is an implementation of the bilinear complementarity algorithm presented in Wilson [7] and in Elken [2]. The HRA code implements the homotopy retraction algorithm as described in [2]. The reader must consult the latter work to be able to formulate a problem correctly and to be cognizant of the conditions under which these algorithms will solve his/her problem.

The algorithms have so many features in common that many of the subroutines in the two programs are identical. In particular, both algorithms begin by solving the linear program

$$\begin{aligned} &\text{maximize} \quad x_{NROW} \\ &\text{subject to} \quad [I|D] \, [\tfrac{t}{x}] = b \, , \qquad (2) \\ &t, x \geq 0 \, . \end{aligned}$$

To solve this linear program, the code LPM1 written by John Tomlin is used. Actually, the subroutines comprising LPM1 are part of both codes, BCA and HRA. LPM1 has been documented in J. Tomlin [4]. LPM1 has been converted into a linear complementarity code, LCPL, which has been documented in J. Tomlin [5] and [6]. Since

LCPL uses many of the subroutines of LPM1 without change, the reader is referred to the works cited for a description of how the matrix $A = [I|D]$ is stored and how transformations are processed.

Next we discuss some important details of the implementation of BCA and HRA.

Source Language

The programs are written entirely in FORTRAN IV for the IBM series 360 and 370 computers. They are WATFIV compatible.

Specification

The main program is executed as a job step. The program input is described in Section II.1. All input occurs in subroutines MAIN, INPUT, and BASINP.

Error Indicators

Error indicators and other diagnostics and messages will be indicated in the documentation for each subroutine. The error indicators for the LPM1 portion of the code are described in [4].

Subroutines

The routines making up BCA and HRA are as follows:

BCA		HRA
MAIN		MAIN
BLCONS		BLCONS
FINDP	. . . .	FINDP
PIVOT	. . . .	PIVOT
UPAKC	. . . .	UPAKC
BSCNG	. . . .	BSCNG
SUPERB	. . . .	SUPERB
RECALC	. . . .	RECALC
ENDPNT		ENDPNT
QUADS	. . . .	QUADS
DSENT	. . . .	DSENT
		GTN
		DERIVG
CONCHK	. . . .	CONCHK
FTN		FTN
DERIV		DERIV
NORM	. . . .	NORM
DECOMP	. . . .	DECOMP
SOLVE	. . . .	SOLVE
SING	. . . .	SING
DEBUG	. . . .	DEBUG

The dotted lines indicate the subroutines which are essentially identical.

The remainder of the subroutines are from LPM1 for both programs.

BLOCK DATA
INPUT
FTRAN
BTRAN
FORMC
TRUEDJ
PRICE
CHUZR
WRETA
SHIFTR
INVERT
UNPACK
SHFTE
CHSOL
UPBETA
NORMAL
ITEROP
UNRAVL
CRASH
BASINP
CLEAR

These subroutines are described in greater detail in the remaining sections of this report.

Program Size

The total length of BCA is 3831 source statements (including comments and common blocks, etc.); HRA is 3772 statements long. LPM1 comprises 1819 statements of each of these.

Array Size

The standard version of the codes requires a blank common array of 133,256 bytes of core. This is dependent on the setting of maximum problem dimensions.

There are eleven labelled common blocks which require 97,448 bytes of core. Only the size of common blocks LNCONS and INDXZ are dependent upon problem dimensions.

Accuracy and Convergence

The linear constraints (1) and (2) are satisfied with a tolerance of ZTOLZE with a default of 1.E - 4. The bilinear constraints (3) are satisfied to a tolerance of TOLFZ; 1.E - 10 is the default.

Convergence is guaranteed in the BCA and HRA when the first IH values of b are chosen appropriately. See Elken [2] Chapters V-VI for a discussion of the convergence properties of these algorithms and suggestions for improvements.

Timing

Using the FORTRAN H compiler with OPT = 2 the codes have solved a problem with 6 consumers and 4 goods with $D \in R^{23\times 20}$ in .45 and .38 seconds for the BCA and HRA, respectively. A problem with 3 consumers and 56 goods and $D \in R^{97\times 107}$ required 4.54 and 6.19 seconds for the BCA and HRA to solve. A description of numerical results for these and several more test problems is in Chapter V of [2].

II. The Bilinear Complementarity Algorithm (BCA)

II.1. Input Requirements

The program input consists of 4 (or 5) segments.

A) Parameter specification relevant to equilibrium problem

B) Parameter specification relevant to the auxiliary linear program

C) Data for the LP in MPS standard form

D) Basis input (optional, see [5])

E) Specification of the C matrix (3). (Initial endowments for the individual consumers.)

We will now deal with these in succession.

A. Program Parameter Input

Parameters are changed from their default value for each problem by means of a FORTRAN "NAMELIST" input from the card reader. The first card must start with &PARM1 in column 2 or later; each card must begin with a blank, and the list must be terminated by &END.

Example

```
columns
 1 2 3
   &PARM1  IH = 3, L = 2, ICECHO = 1, &END
```

The parameters which can be defined by the &PARM1 statement are listed below along with their default values.

REAL *4 Tolerances

TOLFZ (DEFAULT = 1.E - 10)

Termination criterion for the solution of the bilinear equations.

TOLBD (DEFAULT = 1.E - 4)

Feasibility tolerance for the linear constraints.

TOLCV (DEFAULT = 1.E - 5)

Termination criterion for the return to the curve.

INTEGER *4 Parameters

IH Number of consumers (traders). This parameter has no default. It must be specified in the &PARM1 statement.

ICNTRL (DEFAULT = 1)

If ICNTRL.LT.0, then the program behaves just like LPM1 to solve one or more linear programs. This may be useful when one desires to solve some preliminary linear programs to determine initial utility levels. Otherwise, an equilibrium problem will be processed by BCA.

IECHO (DEFAULT = 0)

If IECHO = 1 while ICNTRL.LT.0, the D matrix of (1) will be printed out, ten columns at a time.

ICECHO (DEFAULT = 0)

If .EQ.1, the C-matrix of (3) will be printed.

IPROD (DEFAULT = 0)

If .EQ.1, it is assumed that the consumers own certain shares of the producing firms. Thus, data containing those shares must appear following the LP data in 10F7.4 format.

ITLIME (DEFAULT = 500)

The maximum number of cells which we allow the program to pass through.

KOUTB (DEFAULT = 0)

Unit number of output of the optimal basis for the auxiliary linear program. If KOUTB.EQ.0, no basis will be saved.

REAL *4 Parameters

STPMX (DEFAULT = 100)

The maximum possible step in the direction tangent to the curve.

STPRD (DEFAULT = 0.5)

The factor between 0 and 1 used to reduce the steplength when it exceeds STPMX.

B. Linear Program Parameters

Next, a list containing parameters of importance for the linear programming portion of the code must be included. The beginning of this list is &PARAM. We leave the description of these paramaters to [4] with one exception. The parameter IOBJ contains the row number of the objective row in the D matrix. Since this row is always e_{NROW}^T (2) no flexibility is necessary here so we require that IOBJ be the last row of D. Since we did not want to alter the LPM1 part of the code, the user must know how many rows D has and enter the appropriate value for IOBJ in the parameter list. This rather irritating requirement makes bookkeeping easier in many parts of the code.

Also, the auxiliary linear program (4) is a maximization problem while LPM1 is a minimization code by default. This can be remedied in one of three ways:

1) Let the last row of D be $-e^T_{NROW}$,
2) Set ZSCALE =-1.0 in the &PARAM list, or
3) Set IOBJ = -NROW in the &PARAM list.

Remember to do only one of the alternatives above.

C. Problem Input

The D and b referred to in (1) are read from unit KINP (DEFAULT = 5) in slightly modified "MPS format". The card images required are:

NAME Card

This has "NAME" in columns 1 to 4 and the (up to 8 characters) problem name in columns 15-22. This card is optional but highly desirable.

ROWS Card

Has "ROWS" in columns 1-4.

Row Names

Each row is assigned a (unique) name of up to 8 characters, one per card, in columns 5-12. Embedded blanks are allowed (unlike

MPS). Also row types must be supplied in columns 3 of each card

L: $\leq$ constraint

G: $\geq$ constraint

E: = constraint

N: objective function row .

For the equilibrium problem all constraints will be of type "L" except for the objective function.

COLUMNS Card

Has "COLUMNS" in columns 1-7.

Matrix-Element

The non-zero elements of D are supplied by column. All the elements of a column must be together. Each column is assigned a (unique) name of up to eight characters. The format is:

cols	1-4	5-12	15-22	25-36	40-47	50-61
	blank	column name	row name	element value	second row name (optional)	second element value (optional)

Again embedded blanks are allowed in column names.

RHS Card

Has "RHS" in columns 1-3.

Right Hand Side Elements

The right hand side vector (b) may be given a name, which should be different from any row or column name. The elements are given in the same format as the matrix elements.

ENDATA Card

Has "ENDATA" in columns 1-6. Sample input is shown in Section II.4.

D. LP Basis Input

If KINB is initialized to a nonzero value in &PARAM, a basis is read from unit KINB. If KINB.EQ.KINP, the basis must be entered following the LP problem input, and before the ENDATA card. This basis will be the initial basis for solving the auxiliary LP (2).

NAME Card

Has "NAME" in columns 1-4 and the basis name in columns 15-22. If this name does not agree with the problem name, or the name card is missing, a warning is given.

Basis Cards

For basic columns of D, the column names are given (one per card) in columns 5-12. For basic columns of I (slack variables), the corresponding row name is given in columns 5-12.

ENDATA Card

Has ENDATA in columns 1-6.

E. C Matrix Input

In most problems the data for C will be predominately extracted from the RHS vector b. For this reason we use a rather short, but complicated, form for the input of C.

The data is read from unit KINP directly following the MPS format data. The instructions constructed from the pointers in this data produce a matrix packed into a vector in the same manner that [I|D] is packed (see [6]).

In the current version of the code it is assumed that (if IPROD.NE.0) each consumer I is associated with a fraction SHR(I), which represents his share of each firm in the economy, such that $\sum_{I=1}^{IH}$ SHR(I) = 1.0. If this assumption is not true, the correct C matrix can still be input; however, it requires more work.

The form of the data for inputting the C matrix is as follows

1. One or more cards containing (SHR(I), I = 1, IH) in 10F7.4 format (only if IPROD.NE.0).

2. Several cards describing the column of C associated with consumer 1.
3. A blank card.
4. Information associated with consumer 2, etc.

We now elaborate on the specific requirements of point 2 above. Assume that we are dealing with column I of C.

The first card for column I contains values for the integer variables ISHR, LL, and KK in 3I4 format. ISHR is a control parameter which behaves as follows:

A) If ISHR.EQ.1, the program reads one or more consecutive elements of the RHS corresponding to constraints on one or more of the firms in the economy. These numbers are multiplied by SHR(I), the Ith consumers share of the firm, and stored in C.

B) If ISHR.EQ.0, the program reads one or more consecutive elements of the RHS corresponding to the constraints on consumer I alone.

C) If ISHR.EQ.-1, the program reads one or more elements from the card(s) following in the input stream in 10F7.4 format. These usually correspond to commodities (or firms) which are owned by a number of consumers, but not in any fixed proportion.

LL and KK determine the range of rows of C which will be altered due to this card. If only one row is involved, either KK can be set equal to zero (leave columns 9-12 blank), or set KK equal to LL.

A blank card indicates that the program should move on to the next column of C (the next consumer).

If the problem has been formulated and input correctly, the sum of the columns of C must be equal to the RHS vector (see [2], Section V.3). Examples of input and output for some small problems will be given after the program in Section II.4.

II.2. Main Program

The main program consists of 350 statements which perform a variety of functions: the parameters are initialized and altered by reading the name list PARM1, the subroutines of LPM1 are called to solve the auxiliary linear program, the data is read so that the C matrix can be constructed, the machinery is set up for the bilinear complementarity algorithm to begin, the main path-following subroutine ENDPNT is called, the updates and decisions made necessary by the output from ENDPNT are performed, and the final solution is printed. If options were added so that a number of equilibrium problems could be solved, this main program would undoubtedly become sort of a master subroutine, as NORMAL is for LPM1.

Restrictions relevant to the use of BCA.

1. The number of consumers (IH) must be less than or equal to 10.
2. A must have not more than 350 rows or 400 columns.

Of course these limits can be extended by redimensioning all vectors of size 10, 350 or 400 and allocating more core.

Before describing the subroutines of BCA we will define the variables in the labelled common blocks and some of those in blank common (those of use in the subroutines outside of LPM1).

Variable Glossary

We have specified that almost all real variables are REAL*8 by the statement

```
IMPLICIT  REAL*8  (A-H, O-Z) .
```

Notable exceptions are the initial data in A and C which are stored as REAL*4 variables.

COMMON/LP1/:

PI(1302)   The complete set of dual variables (prices and relative costs).

XX(1302)   The complete set of primal variables, slacks first and then the structural variables in their original order.

COMMON/BLCST/   (Bilinear constraints)

COMMON/BLCST2/

These variables store the current version of the budget surplus function in terms of the superbasic variables (see Elken [2] Section III.3). The functional form is

$$\begin{aligned}&\text{Type 1:}\quad f_\mu = \text{BF1} + \text{D1*PI(MU)} - \text{DIAG(PI(MU))*(F1*XX(NU)} + \text{E1)}\\&\text{Type 2:}\quad f_\nu = \text{BF2} + \text{D2**PI(MU)} - \text{DIAG(XX(NU))*(F2*PI(MU)} + \text{E2)}\ ,\end{aligned}\tag{3}$$

where PI(MU) refers to those components PI(J) of PI such that MU(I) = J for some I = 1, 2, ..., KMU, KMU is the current number of dual superbasic variables, and

$$\text{DIAG(PI(MU))} = \begin{pmatrix} \text{PI(MU(1))} & 0 & \cdots & 0 \\ 0 & \text{PI(MU(2))} & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & \text{PI(MU(KMU)} \end{pmatrix}.$$

XX(NU) and KMU are defined similarly.

Also in COMMON/BLCST/ are C(800), IC(800), and LC(20) which store C in packed form in exactly the same manner as A.

COMMON/LNCONS/

These matrices and vectors allow us to express the basic variables in terms of the superbasic variables. If IH and DIGMA are index sets for the primal and dual variables, respectively, then the relationship is

$$XX(JH(I)) = \sum_{J=1}^{KNU} G1(I,J)*XX(NU(J) + BA(I) \qquad i = 1, \ldots, M$$

$$PI(DIGMA(I)) = \sum_{J=1}^{KMU} G2(I,J) * PI(MU(J)) + BB(I) \qquad I = 1, \ldots, N ,$$

where the arrays are dimensioned G1(350,10), G2(400,10), BA(350), BB(400).

COMMON/INDX1/NUH(10), MUH(10), NU(10), MU(10)

$$NUH(I) = \begin{cases} 0 & \text{if } XX(I) \text{ is not superbasic} \\ J & \text{if } XX(I) \text{ is the Jth superbasic variable } I = 1, \ldots, IH \end{cases}$$

$$MUH(I) = \begin{cases} 0 & \text{if } PI(I) \text{ is not superbasic} \\ J & \text{is } PI(I) \text{ is the Jth dual superbasic variable, } I = 1, \ldots, IH \end{cases}$$

NU(I) = J if the Ith superbasic variable is XX(J)

MU(I) = J if the Ith dual superbasic variable is PI(J).

Note that NUH(NU(I)) = I.

COMMON/INDX2/JH,DIGMA,KINBAS,IDBAS

These arrays correcpond to σ, $\bar{\sigma}$, γ and $\bar{\gamma}$ in the description of the algorithm in Chapter III of [2].

JH(I) = J if XX(J) is basic and pivots on row I.

DIGMA(I) = J if PI(J) is dual basic and corresponds to the Ith row of G2.

$$\text{KINBAS(I)} = \begin{cases} 0 \text{ if XX(I) is non-basic} \\ \text{J if XX(I) is basic and pivots on row J.} \end{cases}$$

$$\text{IDBAS(I)} = \begin{cases} 0 \text{ if PI(I) is not dual basic} \\ \text{j if PI(I) is dual basic and corresponds to the Jth row of G2.} \end{cases}$$

COMMON/SCAL/

BT = A scalar that helps to define the normal hyperplane subproblem in ENDPNT.

JJ = The index of the constraint in G1 or G2 which defines the exiting facet for the current cell.

MFLAG = A flag which defines the type of jacobian which must be calculated.

DD1 = The number of bilinear constraints which are currently at zero.

P = The index of the pivot column determined by subroutine FINDP.

PD = The flag which determines whether the initial facet corresponds to a primal (PD = 1), a dual (PD = -1) variable, or a bilinear constraint (PD = 0) at zero.

MPD = Identical to PD except that if refers to the final facet of the currant cell.

$$\text{INEQ} = \begin{cases} \text{1, if the bilinear inequality is the last of the Type 1 constraints.} \\ \text{2, if the bilinear inequality is the last of the Type 2 constraints see (1).} \end{cases}$$

KFUN = Counts the number of functionals evaluated in (1).

KJAC = The number of partial derivatives of the functions defined by (1) which are calculated during the course of the algorithm.

COMMON/DIM/ (DIMENSIONS)

IH = The number of households.

M = Number of rows in the problem.

N = Number of structural variables in the problem.

DD = Current number of superbasic variables (= DD1 + 1).

KMU = Current number of primal superbasics.

KNU = Current number of dual superbasics (DD = KMU + KNU)

$$\text{BL1} = \begin{cases} \text{KMU - 1, if INEQ = 1} \\ \text{KMU, otherwise} \end{cases}$$

$$\text{BL2} = \begin{cases} \text{KNU - 1, if INEQ = 2} \\ \text{KNU, otherwise} \end{cases}$$

MP1 = M + 1

NM = N + M

COMMON/INT/

IPS(30) A vector storing the permutations for a LU decomposition linear equation solver used in conjunction with Newton's method.

KDET The sign of the determinant of the current jacobian of f.

KOUNT The number of times ENDPNT has been called.

ISING A flag which is 1 when the current jacobian is singular, 0 otherwise.

COMMON/TOLER/TOLFZ, TOLBD, TOLCV, ICNTRL, IECHO

These parameters are defined above.

II.3. Subroutines of BCA

(2) SUBROUTINE BLCONS: Calculates the coefficients of the first DD bilinear constraints. DD1 of these are constraints which define the curve. The last is an inequality which is not yet binding, but which must be the next one to become binding.

The calculations are motivated by the fact that the bilinear functions

$$\sum_{j=1}^{M} C(I,J)*PI(J) - PI(I)*XX(I), \quad I = 1, \ldots, DD$$

can be reduced to functions of the superbasic variables by using (6). The position of DD in the index sets MU and NU determines BL1, BL2, INP, INQ, and INEQ.

(3) SUBROUTINE FINDP (PD1, IS, P): Finds the element of largest absolute value in the row determined by (PD1, IS) and stores the index of the variable corresponding to that element in P. When we.say the row determined by (PD1, IS) we mean $G1(IS,\cdot)$ if $PD1 = 1$, and $G2(IS,\cdot)$ if $PD1 = -1$.

(4) SUBROUTINE PIVOT (SS,RR): If variable XX(SS) is entering the basis and variable XX(RR) is leaving the basis, the columns of G1 and G2 and the columns BA and BB must be updated. This is accomplished by a simple pivot. The details for this pivot are contained in Section III.5 of [2].

(5) SUBROUTINE UNPACKC(II): A call to this subroutine causes the IIth column of C to be unpacked and stored in the M-vector Y.

(6) SUBROUTINE BSCNG(S,R): This subroutine updates the index sets JH, KINBAS, DIGMA, and IDBAS when variable S replaces variable R in the primal basis.

(7) SUBROUTINE SUPERB(KEY, PD1, IS, PD2,JS): Revises the index sets defining the superbasic variables. It also adds or removes columns of G1 and G2 depending on the values of the parameters

$$KEY = \begin{cases} 0, \text{ if columns are both added and removed,} \\ 1, \text{ if columns are only added,} \\ 2, \text{ if columns are only removed.} \end{cases}$$

(PD1,IS) determine the column to be added.

(PD2,JS) determine the column to be removed.

The calculations iuvolved in calculating the column to be added are described in [2] Section III.3.

(8) SUBROUTINE RECALC: The parameter INVFRQ of LPM1 specifies how frequently the product form basis is to be reinverted. Each time the basis is reinverted while the BCA portion of the code is in effect, G1, G2, and BA, BB are recomputed using the new basis inverse. The same thing could be accomplished by DD calls to SUPERB with KEY = 1, and (PD1, IS) appropriately specified.

(9) SUBROUTINE ENDPNT (JS, PD1, IS, NET): implements the path-following algorithm described in Section III.4 [2]. The initial point is in a facet of the cell defined by (6) and the bilinear inequality; we call this the initial facet. The algorithm follows the curve defined by the DD1 bilinear constraints and the DD superbasic variables. The objective is to follow the curve until it intersects another facet of the cell and to find the point where that intersection occurs. We call that point the endpoint which is contained in the final facet. The initial facet is determined by the equation described by (PD1, JS). The final facet is determined by (PD1, IS). NET counts the number of Newton iterations which were required in the various subproblems of the algorithms referred to above.

A common block is referred to for the first time in this subroutine: COMMON/NEWT/H(10, 11), X(10) Z(10), ACC(2, 10). The following vectors are used in this routine: U(3), V1(10), (V2(10) F(10), DOT(4), RHS(10), UL(10, 10).

H(10, 11) stores the jacobian for the various Newton subproblems

X(10) contains the superbasic variables as X = (PI(MU), XX(NU))

Z(10) is the correction to X supplied by the Newton method, $Z = H^{-1}F$

ACC(2, 10) sotres the two vectors which determine the current linear approximation to the curve. $f^{-1}(0)$

U(3) stores the coefficients for any quadratics that have to be solved

V1(10) holds the gradient of the functional defining the initial facet

V2(10) contains a vector in the null space of H, a tangent to $f^{-1}(0)$

F(10) holds the values of f and any other functional involved in the current subproblem

DOT(4) a collection of accumulators

RHS(10) a work space for finding the tangent vector by solving a linear system; RHS = H(·, ICX).

UL(10, 10) stores the LU decomposition of the current jacobian matrix

Subroutines called: QUADS, DSENT, CONCHK, FTN, DERIV NORM, DECOMP, SOLVE.

It should be noted that the Newton's method being implemented is a modified Newton's method with descent. The iteration is

$$X^{k+1} = X^{k} - \alpha(f'(x^{k})^{-1} f(x^{k})) \qquad k = 0, 1, \ldots$$

where α is the first scalar of $\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots\}$ such that

$$\|f(X^{k+1})\| < \|f(X^{k})\| \ .$$

The basic loop is something like the following

```
DO 100 K = 1, 15
      CALL FTN(F, X)                         calculate  f(X)
      CALL NORM (F, S1, DD)                  set  S1: = ‖f(X)‖
      IF (S1.LT.TOLCV) GO to 800             terminate if  S1 < 10^-5
      CALL DERIV                             calculate  H: = f'(X)
      CALL DECOMP(DD, H, UL)                 decompose  H = UL
      CALL SOLVE(DD, UL, F, Z)               solve by back substitution
                                               UL*Z = F
      ALPHA = 1.0
      CALL DSENT (ALPHA, S1)                 implements descent on the
                                             norm of  f.
      DO 95 I = 1, DD
         X(I) = X(I) - ALPHA*Z(I)
 95   CONTINUE
100   CONTINUE
 C    TAKE CORRECTIVE ACTION IF NEWTONS
 C    METHOD HAS NOT TERMINATED WITHIN
 C    15 ITERATIONS
```

(10) SUBROUTINE QUADS (U, IMAG, ALPHA, BETA): finds both real roots of the quadratic

$$U(1) \cdot \alpha^2 + U(2) \cdot \alpha + U(3) = 0 \quad .$$

If there are no real roots, the flag IMAG is set equal to 1. If there is a double root or if U(1) = 0, the root is stored in both ALPHA and BETA. If there are two real roots ALPHA stores the smaller root and BETA the larger.

(11) SUBROUTINE CONCHK (GMIN, KGMIN, MP2): checks whether the constraints of (1) are satisfied at the point defined by the current superbasic values. The basic variables are evaluated in terms of the superbasic variables as suggested by (4). GMIN stores the minimum value of these variables and (MP2, KGMIN) indicate which variable attains this minimum. The theory tells us that we only have to check the current bilinear inequality among the bilinear constraints described in (3). If this value is less than GMIN, MP2 is set equal to zero.

(12) SUBROUTINE FTN(F, X): evaluates the first DD1 bilinear functionals as they were defined in BLCONS (3). The last component F(DD) stores a different function value depending upon which subproblem is being solved in ENDPNT.

If MFLAG = 0, then F(DD) = 0.

If MFLAG = 1, then F(DD) contains the value of the functional specified by (MPD, JJ).

If MFLAG = 2, then F(DD) = X(JJ).

If MFLAG = 3, then F(DD) = ⟨ACC(2,·), X⟩ - BT (the normal hyperplane to ACC(2,·), if F(DD) = 0).

(13) SUBROUTINE DERIV: computes the exact jacobian of the function calculated in f. The form of the jacobian is as follows

$$H = \begin{array}{cc} \text{PI(MU)} & \text{XX(NU)} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} \text{D1} & \text{DIAG(PI(MU))*F1} \\ \hline \text{D2} & \text{-DIAG(F2*(PI(MU) + E2)} \end{array}\right]} \end{array} - \begin{array}{cc} \text{PI(MU)} & \text{XX(MU)} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} \text{DIAG(F1*XX(NU) + E1)} & 0 \\ \hline \text{DIAG(XX(NU))*F2} & 0 \end{array}\right]} \end{array}$$

The last row of H contains the gradient of the last functional of F as described above.

(14) SUBROUTINE DSENT (ALPHA, PNORM): Implements descent on the norm of f in Newton's method as described above. ALPHA is cut in half until $\|f(x^{k+1})\| < \text{PNORM} = \|f(x^k)\|$ or until ALPHA $\leq 10^{-5}$.

(15) SUBROUTINE DECOMP (NN, A, UL): (Borrowed from Forsythe [3]. The reader should look there for a complete description.) Implements a Gaussian elimination scheme with partial pivoting to produce a LU decomposition of the matrix A. A few statements were added to this subroutine so that KDET would change sign every time a row interchange was performed. KDET will eventually be the sign of det A (see Elken [1977a], sec. III.2).

(16) SUBROUTINE SOLVE (NN, UL, B, X): is also borrowed from Forsythe [3] with the additions of statements which change the sign of KDET every time UL(I, I) < 0. SOLVE uses backsubstitution to solve the system L*U*X = B. L and U are a lower and upper triangular matrix which are stored in the square matrix UL.

At the end of SOLVE, KDET is equal to sgn(det A).

(17) SUBROUTINE SING(IWHY): is also borrowed from [3]. It is called from DECOMP and SOLVE to print messages concerning the singularity of A depending on the parameter IWHY.

(18) SUBROUTINE DEBUG(MODE): prints information which may be useful in debugging the program when changes are made. The information which is printed depends upon the value of MODE.

MODE = 1: Prints the time left in this job step by calling the system subroutine LEFT1A(TIME) where TIME is a REAL*4 variable. When running this program in WATFIV or on a system other than SLAC, this portion of code should be removed.

MODE = 2: The arrays BA, G1, BB, and G2 will be printed by rows.

MODE = 3: The index sets defining the primal basis, JH and KINBAS will be printed.

MODE = 4: Information about the location and value of the first IH primal and dual variables will be given (KNU, KMU, NVH, MUH, (XX(I), PI(I), I = 1, ..., IH)).

MODE = 5: The coefficients of the bilinear constraints will be printed: BF1, BF2, D1, D2, E1, E2, F1, and F2.

MODE = 6: A variety of variables will be printed out which are of use in debugging ENDPNT: PD, KMU, KNU, INEQ, MUH, NUH, JH, DIGMA, PI and XX.

The remaining 1815 lines of BCA contain the subroutines which comprise LPM1. The reader is referred to [4], [5], and [6] for documentation relating to them.

II.4. Sample Problems

The sample input and output are given below for two small problems. The computational results for these problems are given in [2] Chapter V.

For the reader to learn how to formulate an equilibrium problem so that it can be input and solved by BCA, (or HRA) it is necessary to read Sections II.2 and II.3 of [2]. We will discuss the definitions of D, b and C of (1) for a small problem here.

The problem devised by Mas Colell (Wilson [7] is concerned with three consumers and two goods. By using the theory of Section II.2 of Elken [2] we see that solving the equilibrium problem devised by Mas Colell is equivalent to solving (1) with

$$D = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ .5 & 1. & .25 & 1. \\ 1. & .5 & .20 & 1. \end{bmatrix}, \qquad b = \begin{bmatrix} -0.9 \\ -0.95 \\ -3.92 \\ 3.0 \\ 3.0 \end{bmatrix},$$

and

$$C = \begin{bmatrix} -0.9 & 0 & 0 \\ 0 & -0.95 & 0 \\ 0 & 0 & -3.92 \\ 1. & 1. & 1. \\ 1. & 1. & 1. \end{bmatrix} .$$

Notice that the sum of the columns of C is b. This will always be the case. We are almost ready to type the data cards for BCA, but we have to add a row to D to represent the objective row and assign names to the rows and columns. Below is our choice of names.

$$D = \begin{array}{l|cccc|} & \text{ACT1} & \text{ACT2} & \text{ACT3} & \text{MEXP} \\ \text{UTIL1} & -1 & & & \\ \text{UTIL2} & & -1 & & \\ \text{UTIL3} & & & -1 & \\ \text{GOOD1} & .5 & 1. & .25 & 1. \\ \text{GOOD2} & 1. & .5 & .20 & 1. \\ \text{OBJ} & & & & 1. \end{array}$$

We only enter the nonzero values to emphasize the fact that only the nonzero elements of D and b need be entered.

In the sample input below (Figure 1) we specify that we want

three consumers	(IH = 3) ,
two goods	(L = 2) ,
no production	(IPROD = 0) ,

and a

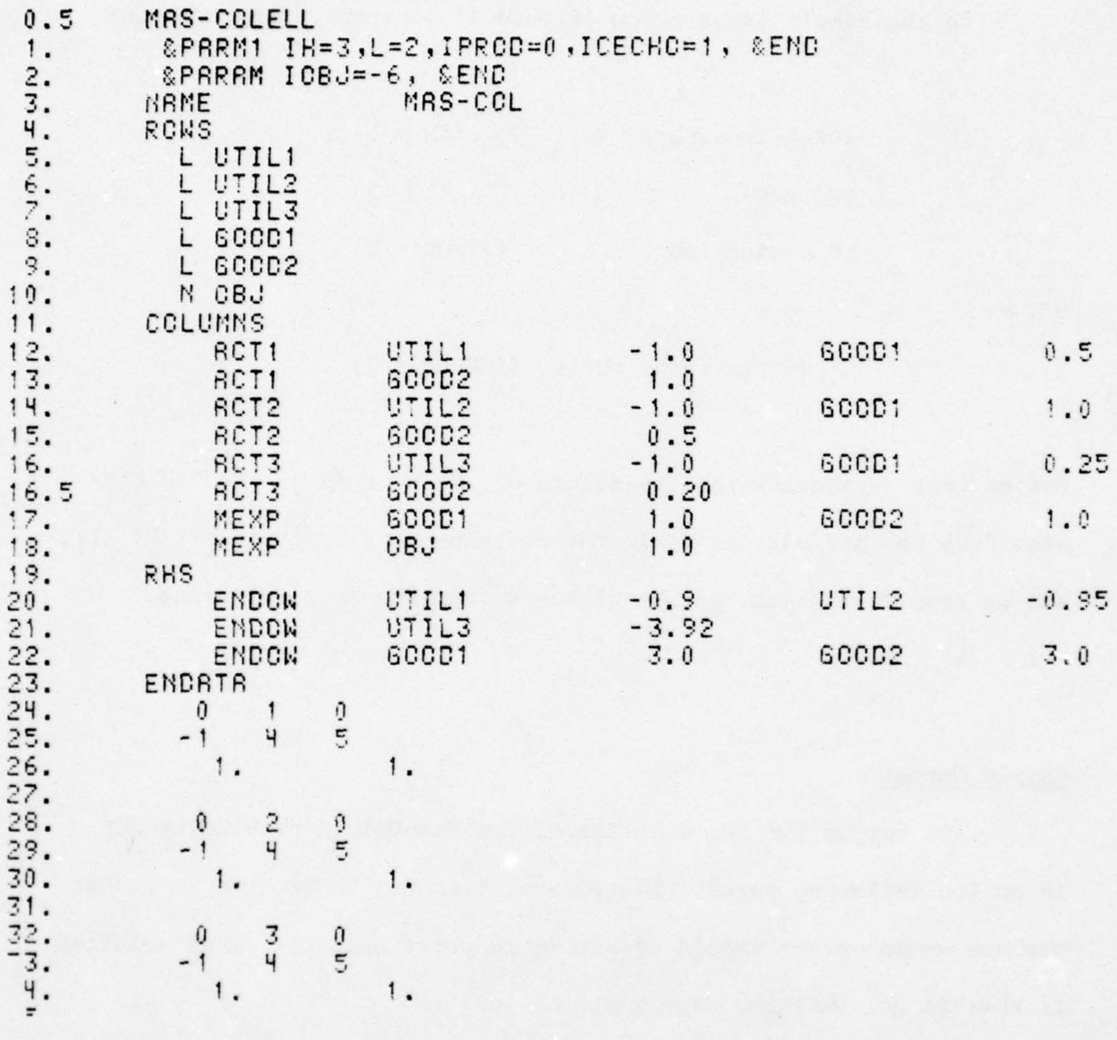
C matrix input check (ICECHO = 1) .

Notice that we specify the Ith column of C by reading the Ith element from b and placing it in the corresponding position of C(·,I), and we read in the 4th and 5th elements from the card following '-1 4 5'.

Sample Output

The output for the solution of the Mas-Colell problem by BCA is on the following pages. The current form of the program is rather verbose -- an option should be put in to print only the final solution if that is all that the user wants to see.

The first page of output shows the values of all parameters, gives the problem name and the output concerning the solution of the auxilliary linear program. Messages concerning time can be ignored in this printout because no timing routine was being called.



```
MAS-COLELL
 &PARM1 IH=3,L=2,IPROD=0,ICECHO=1, &END
 &PARAM IOBJ=-6, &END
NAME               MAS-COL
ROWS
  L UTIL1
  L UTIL2
  L UTIL3
  L GOOD1
  L GOOD2
  N OBJ
COLUMNS
     ACT1       UTIL1          -1.0         GOOD1          0.5
     ACT1       GOOD2           1.0
     ACT2       UTIL2          -1.0         GOOD1          1.0
     ACT2       GOOD2           0.5
     ACT3       UTIL3          -1.0         GOOD1          0.25
     ACT3       GOOD2           0.20
     MEXP       GOOD1           1.0         GOOD2          1.0
     MEXP       OBJ             1.0
RHS
     ENDOW      UTIL1          -0.9         UTIL2         -0.95
     ENDOW      UTIL3          -3.92
     ENDOW      GOOD1           3.0         GOOD2          3.0
ENDATA
    0    1    0
   -1    4    5
     1.         1.

    0    2    0
   -1    4    5
     1.         1.

    0    3    0
   -1    4    5
     1.         1.
```

Figure 1

The following pages give information concerning the behavior of the bilinear complementarity algorithm, and finally, the solution. As a measure of the work done by the algorithm, the number of calls to subroutine ENDPNT, the number of scalar function evaluations, and the number of IX × IH jacobian evaluations are printed. The solution is printed in a common format for linear programs:

JH(I) The names of the primal variables which may be non-zero in the equilibrium solution

VALUE gives the value of these variables. The value of MEXP should always be near zero in an equilibrium.

ROW NAMES is self-explanatory

PI(I) gives the value of the dual multiplier associated with that row. Those PI's associated with commodity balance rows are the usual "equilibrium prices."

RHS gives the value of the original RHS vector (b).

.18 sec

```
&PARM1
TOLFZ=  .1000000000000000000D-09,TOLBD=  .1000000000000000001D-03,TOLCV=  .9999999999999999997D-05,ICNTRL=
         0,STPMX=  10.0000000000000000      ,STPRD=  .5000000000000000000      ,ICFCHO=              1,IH=           3
2,IPROD=
         0,KOLTE=              0,ITLIME=          100
&END
  IECHO,IH,ICECHC    0    3    1
&PARAM
ZTOLZE=  .100000005E-03,ZTOLPV=  .100000034E-05,ZTCOST=  .100000005E-03,ZSCALF= -1.00000000      ,NRMAX=
   2000.
NEMAX=        9000,IALG=              0,NDEGI=              1,NDUAL=            999,NIPIW=        99999,KINP=             5,IFB/
0,IFCRSH=
         1,IOBJ=              6,ITCH=              30,INVFRQ=              30,ITRLIM=         99999,ZTOLRP=  .999999975E-05,Z1
)00016E-01.
ZTETA=  .100000043E-11,KINB=                0,ZTCLDA=  .100000008E-08
&END
PROBLEM   MAS-COL
                                                                                                          1,IFCHO=
                                                                                                      ,L=
PROBLEM STATISTICS
     6 ROWS
     4 STRUCTURAL COLUMNS
    12 NON-ZERO ELEMENTS
DENSITY = 0.50000
 THE TIME LEFT IS NOW   1.000000 SEC.                                                              350,NTMAX=
INVERT STATISTICS
   14 NONZ IN BASIS                                                                                IS=
    4 STRUCTURAL COLUMNS IN BASIS
    3 VECTORS ABOVE BUMP                                                                           TOLRI=   .100
    3 VECTORS BELOW BUMP
L:       9 NONZ      3 ETAS
U:       3 NONZ     -1 ETAS
TOTALS:      8 OFF DIAG NONZ     4 ETAS
RELATIVE ERROR  IN X =      0.63725E-16

ITCOUNT   STATUS     NINF     OBJ VALUE         VECIN      VECOUT     NNEGDJ             DJ                 NETA   NELEM
    1      I          1        0.22100002       GOOD2      GOOD1         2          -1.00000000              4      12
    1      O          0       -0.61999995       GOOD2      GOOD1         0           0.25000000              5      15
 MAS-COL

JH(I)                      VALUE         ROW NAMES           PI(I)           RHS
ACT1                     0.90000000       UTIL1            0.50000000     -0.90000000
ACT2                     0.95000000       UTIL2            1.00000000     -0.95000000
ACT3                     3.92000000       UTIL3            0.25000000     -3.92000000
GOOD2                    0.22100005       GOOD1            1.00000000      3.00000000
MEXP                     0.62000000       GOOD2            0.0             3.00000000
OBJ                     -0.62000000       OBJ             -1.00000000      0.0
 THE TIME LEFT IS NOW   1.000000 SEC.


      STRUCTURAL COLUMNS OF MATRIX.
```

```
COLUMNS      1 THROUGH      3
    -0.9000      0.0         0.0
     0.0        -0.9500      0.0
     0.0         0.0        -3.9200
     1.0000      1.0000      1.0000
     1.0000      1.0000      1.0000
     0.0         0.0         0.0

--------------------------------------------------
THE     1TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
            A2(*U)+                  A3
           1.0000000             0.0
STEPLENGTH=             0.442000  CONSTRAINT TYPE:(  1,   4)
 ITERATION  0 X=         0.44200
NORM(F(X))=    0.0           F=    0.0
 AFTER    0 NEWTON ITERATIONS.
 THE    5TH PRIMAL VARIABLE WENT TO ZERO.
 VECIN: UTIL1      VECOUT: GCCD2

--------------------------------------------------
THE     2TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
            A2(*U)+                  A3
           1.0000000             0.5000000
STEPLENGTH=             0.245156  CONSTRAINT TYPE:(  0,   7)
 ITERATION  0 X=         0.74516
NORM(F(X))=    0.0           F=    0.0
 AFTER    0 NEWTON ITERATIONS.

 THE BILINEAR CONSTRAINT HIT ZERO.
THERE ARE NOW ,  1 BUDGET SURPLUSES AT ZERO.

--------------------------------------------------
THE     3TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
            A2(*U)+                  A3
          -0.4854442             0.7451564
           0.8742676             0.0
STEPLENGTH=             0.304255  CONSTRAINT TYPE:(  1,   5)
 ITERATION   0 X=        0.59746   0.26600
NORM(F(X))=    0.039288  F=    0.039288 -0.000000
 ITERATION   1 X=        0.62189   0.26600
NORM(F(X))=    0.0           F=    0.0         0.0
           TANGENT
THE ~~QUADRATIC~~ PICKED THE WRONG CONSTRAINT.
 THE MOST INFEASIBLE CONSTRAINT IS TYPE    0  NUMBER     5
WITH A VALUE OF           -0.067781
 ITERATION   0 X=        0.80332  -0.12552
NORM(F(X))=    0.426211  F=    0.022773  0.425603
 ITERATION   1 X=        0.58738   0.22983
NORM(F(X))=    0.108519  F=    0.076734 -0.076734
 ITERATION   2 X=        0.64148   0.21569
NORM(F(X))=    0.001082  F=    0.000765 -0.000765
 ITERATION   3 X=        0.64203   0.21555
NORM(F(X))=    0.000000  F=    0.000000 -0.000000
```

```
SUPERBASIC INFO: KNU=   1   KMU=    1
NUH(I)=     0  1  0
MUH(I)=     1  0  0
  T    =     0.657548     0.215548     0.0
LAMBDA=     0.642035     0.857965     0.235797
 THE TIME LEFT IS NOW   1.000000 SEC.
 AFTER    4 NEWTON ITERATIONS.

 THE BILINEAR CONSTRAINT HIT ZERO.
THERE ARE NOW ,  2 BUDGET SURPLUSES AT ZERO.

-----------------------------------------------
THE     4TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
         A2(*U)+                 A3
        -0.0263976            0.6420346
        -0.0358613            0.2155483
         0.9990081            0.0
STEPLENGTH=            0.079954  CONSTRAINT TYPE:(   0,   7)
 ITERATION   0 X=     0.63992    0.21268    0.07988
NORM(F(X))=    0.055999   F=    0.000011   0.000006   0.055999
 ITERATION   1 X=     0.63394    0.20460    0.30701
NORM(F(X))=    0.000169   F=    0.000088   0.000048 -0.000136
 ITERATION   2 X=     0.63397    0.20470    0.30650
NORM(F(X))=    0.000000   F=   -0.000000   0.000000 -0.000000
 AFTER    2 NEWTON ITERATIONS.

 THE BILINEAR CONSTRAINT HIT ZERO.
THERE ARE NOW ,  3 BUDGET SURPLUSES AT ZERO.


AFTER    4 ENDPNT CALLS, WE HAVE EQUILIBRIUM.

TUTALS:   SCALAR FUNCTION CALLS=        37
            JACOBIAN EVALUATIONS=        4

 THE TIME LEFT IS NOW  1.000000 SEC.
 UTILITY LEVELS:          1.5773504        1.1547005        4.2264982
 PRICES:        0.633975     0.866025     0.236603
 MAS-COL


JH(I)                    VALUE            ROW NAMES             PI(I)             RHS
ACT1                    1.57735041          UTIL1              0.63397455       -0.90000000
ACT2                    1.15470049          UTIL2              0.86602545       -0.95000000
ACT3                    4.22649817          UTIL3              0.23660254       -3.92000000
UTIL1                   0.67735041          GOOD1              0.73205090        3.00000000
MEXP                   -0.00000023          GOOD2              0.26794910        3.00000000
OBJ                     0.00000023          OBJ               -1.00000000        0.0
```

Sample Problem: Whisman

Next we give a sample input and output for a problem formulated by Al Whisman (Wilson [1976]). The problem involves four traders and three goods. Each trader is concerned with two to four activities. To describe the problem we give only the D, b, and C matrices of problem (1):

$$D = \begin{bmatrix} -1 & -1 & & & & & & & & & \\ & & -1 & -1 & & & & & & & \\ & & & & -1 & -1 & -1 & -1 & & & \\ & & & & & & & & -1 & -1 & \\ 1 & 2 & 4 & 3 & & 1 & & 3 & 1 & 2 & 1 \\ 2 & 2 & 3 & 2 & 1 & 2 & 3 & & 1 & & 1 \\ 3 & 2 & 1 & 2 & & & 4 & 3 & & & 1 \\ & & & & & & & & & & 1 \end{bmatrix},$$

$$b^T = [-.48 \quad -.32 \quad -5.1 \quad -.95 \quad 5 \quad 11 \quad 9],$$

$$C = \begin{bmatrix} -.48 & & & \\ & -.32 & & \\ & & -.51 & \\ & & & -.95 \\ 2 & 1 & 1 & 1 \\ 2 & 3 & 5 & 1 \\ 1 & 3 & 4 & 1 \end{bmatrix}.$$

Again, only the non-zero elements of the matrices are given.

The input follows. The first card, again, is just a header; it is not to be included in the input stream.

The sample output for the Whisman problem also follows, page 40.

```
WHISMAN
 &PARM1 IH=4,L=3,IPROD=0,ICECHO=1, &END
 &PARAM IOBJ=-8, &END
NAME           WHISMAN
ROWS
  L UTIL1
  L UTIL2
  L UTIL3
  L UTIL4
  L GOOD1
  L GOOD2
  L GOOD3
  N OBJ
COLUMNS
    C1AC1     UTIL1          -1.0       GOOD1         1.0
    C1AC1     GOOD2           2.0       GOOD3         3.0
    C1AC2     UTIL1          -1.0       GOOD1         2.0
    C1AC2     GOOD2           2.0       GOOD3         2.0
    C2AC1     UTIL2          -1.0       GOOD1         4.0
    C2AC1     GOOD2           3.        GOOD3         1.
    C2AC2     UTIL2          -1.0       GOOD1         3.
    C2AC2     GOOD2           2.        GOOD3         2.
    C3AC1     UTIL3          -1.        GOOD2         1.0
    C3AC2     UTIL3          -1.        GOOD1         1.0
    C3AC2     GOOD2           2.
    C3AC3     UTIL3          -1.        GOOD2         3.
    C3AC3     GOOD3           4.
    C3AC4     UTIL3          -1.0       GOOD1         3.
    C3AC4     GOOD3           3.0
    C4AC1     UTIL4          -1.0       GOOD1         1.
    C4AC1     GOOD2           1.0
    C4AC2     UTIL4          -1.        GOOD1         2.
    MEXP      GOOD1           1.        GOOD2         1.
    MEXP      GOOD3           1.        OBJ           1.
RHS
    ENDOW     UTIL1          -0.48      UTIL2        -0.32
    ENDOW     UTIL3          -5.1       UTIL4        -0.95
    ENDOW     GOOD1           5.        GOOD2        11.0
    ENDOW     GOOD3           9.
ENDATA
   0    1    0
  -1    5    7
    2.          2.          1.

   0    2    0
  -1    5    7
    1.          3.          3.

   0    3
  -1    5    7
    1.          5.          4.

   0    4
  -1    5    7
    1.          1.         1.

```

```
          0,STPMX=  10.000000000000000      ,STPRD=  .500000000000000000      ,ICECHO=             1,IH=          4
3,IPROD=
          0,KOUTB=              0,ITLIME=         100
&END
 IECHO,IH,ICECHO    0    4    1
&PARAM
ZTOLZE=  .100000005E-03,ZTOLPV=  .100000034E-05,ZTCOST=  .100000005E-03,ZSCALE= -1.00000000     ,NRMAX=
    2000,
NEMAX=        8000,IALG=             0,NDEGI=             1,NDUAL=           999,NIPIW=        99999,KINP=           5,IFP
0,IFCRSH=
         1,IOBJ=            8,ITCH=             30,INVFRQ=            30,ITRLIM=        99999,ZTOLRP=  .999999975E-05,7
0000016E-01,
ZTETA=  .100000043E-11,KINB=              0,ZTCLDA=  .100000008E-08
&END
PROBLEM  WHISMAN


PROBLEM STATISTICS                                                                                                   ,L=
     8 ROWS
    11 STRUCTURAL COLUMNS
    36 NON-ZERO ELEMENTS
DENSITY = 0.40909
 THE TIME LEFT IS NOW   1.000000 SEC.                                                                               350,NTMAX=
INVERT STATISTICS
   20 NONZ IN BASIS                                                                                                  AS=
    5 STRUCTURAL COLUMNS IN BASIS
    2 VECTORS ABOVE BUMP                                                                                             TOLRI=   .100
    6 VECTORS BELOW BUMP
L:     8 NONZ      2 ETAS
U:     9 NONZ      3 ETAS
TOTALS:    12 OFF DIAG NONZ       5 ETAS
RELATIVE ERROR IN X =     0.0

ITCOUNT  STATUS      NINF      OBJ VALUE        VECIN      VECOUT      NNEGDJ              DJ              NETA    NELEM
    1      I           2       9.15999889      C3AC3      MEXP            5           -6.00000000           5        17
    2      I           1       0.58999997      C3AC4      GOOD2           4           -2.50000000           6        22
    3      F           0      -0.98874998      MEXP       GOOD1           1           -1.00000000           7        26
    4      F           0      -1.54874992      C2AC2      C2AC1           3           -1.74999905           8        31
    5      F           0      -2.60999966      GOOD3      C3AC4           2           -0.37500000           9        37
    5      O           0      -2.60999966      GOOD3      C3AC4           0            0.0                 10        42
 WHISMAN

JH(I)                   VALUE           ROW NAMES           PI(I)              RHS
C1AC1                  0.48000000        UTIL1            1.00000000        -0.48000000
C2AC2                  0.32000000        UTIL2            3.00000000        -0.32000000
C3AC1                  4.73000000        UTIL3            0.0               -5.10000000
C4AC1                  0.95000000        UTIL4            1.00000000        -0.95000000
MEXP                   2.61000000        GOOD1            1.00000000         5.00000000
GOOD3                  2.83000000        GOOD2            0.0               11.00000000
C3AC3                  0.37000000        GOOD3            0.0                9.00000000
OBJ                   -2.61000000        OBJ             -1.00000000         0.0
 THE TIME LEFT IS NOW   1.000000 SEC.



    STRUCTURAL COLUMNS OF MATRIX.
```

```
COLUMNS     1 THROUGH     4
    -0.4800     0.0         0.0         0.0
     0.0       -0.3200      0.0         0.0
     0.0        0.0        -5.1000      0.0
     0.0        0.0         0.0        -0.9500
     2.0000     1.0000      1.0000      1.0000
     2.0000     3.0000      5.0000      1.0000
     1.0000     3.0000      4.0000      1.0000
     0.0        0.0         0.0         0.0

------------------------------------------------
THE     1TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TC THE CURVE IS:
         A2(*U)+                 A3
       1.0000000            0.0
STEPLENGTH=          0.740000  CONSTRAINT TYPE:(  1.   7)
 ITERATION  0 X=      0.74000
NORM(F(X))=    0.0          F=    0.0
 AFTER    0 NEWTON ITERATIONS.
 THE  15TH PRIMAL VARIABLE WENT TC ZERO.
 VECIN: UTIL1       VECOUT: C3AC3

------------------------------------------------
THE     2TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TC THE CURVE IS:
         A2(*U)+                 A3
       1.0000000            1.0000000
STEPLENGTH=          0.500000  CONSTRAINT TYPE:( -1. 11)
 ITERATION  0 X=      1.50000
NORM(F(X))=    0.0          F=    0.0
 AFTER    0 NEWTON ITERATIONS.
 THE  18TH DUAL VARIABLE WENT TO ZERO.
 VECIN: C4AC2       VECOUT: UTIL1

------------------------------------------------
THE     3TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TC THE CURVE IS:
         A2(*U)+                 A3
       1.0000000            0.7400000
STEPLENGTH=          0.113333  CONSTRAINT TYPE:(  0.   9)
 ITERATION  0 X=      0.85333
NORM(F(X))=    0.0          F=    0.0


SUPERBASIC INFO: KNU=   1   KMU=    0
NUH(I)=    1   0   0   0
MUH(I)=    0   0   0   0
 T    =     0.853333     0.0           0.0           0.0
LAMBDA=     1.500000     2.500000      0.500000      1.000000
 THE TIME LEFT IS NOW  1.000000 SEC.
 AFTER    0 NEWTON ITERATIONS.

 THE BILINEAR CONSTRAINT HIT ZERO.
THERE ARE NOW ,  1 BUDGET SURPLUSES AT ZERO.
```

```
--------------------------------------------------
THE     4TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
          A2(*U)+                  A3
          0.0                  0.8533334
          1.0000000            0.0
STEPLENGTH=           0.113333  CONSTRAINT TYPE:(   1,   7)
 ITERATION  0 X=       0.85333    0.11333
NORM(F(X))=    0.0       F=      0.0         0.0
 AFTER    0 NEWTON ITERATIONS,
 THE  13TH PRIMAL VARIABLE WENT TO ZERO.
 VECIN: UTIL1      VECOUT: C4AC2

--------------------------------------------------
THE     5TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
          A2(*U)+                  A3
        -0.7474093             1.5000000
         0.6643638             0.1133334
STEPLENGTH=           0.533091  CONSTRAINT TYPE:(   1,   5)
 ITERATION  0 X=       1.10156    0.46750
NORM(F(X))=    0.141113  F=    0.141113  0.000000
 ITERATION  1 X=       1.18519    0.46750
NORM(F(X))=    0.000000  F=    0.000000  0.0

THE QUADRATIC PICKED THE WRONG CONSTRAINT.
 THE MOST INFEASIBLE CONSTRAINT IS TYPE    0  NUMBER     5
WITH A VALUE OF        -0.846296
 ITERATION  0 X=       1.82037   -0.24708
NORM(F(X))=    2.492337  F=    0.228929  2.481801
 ITERATION  1 X=       1.13853    0.24310
NORM(F(X))=    0.472660  F=    0.334221 -0.334221
 ITERATION  2 X=       1.32670    0.29484
NORM(F(X))=    0.013770  F=   -0.009737  0.009737
 ITERATION  3 X=       1.32152    0.29342
NORM(F(X))=    0.000010  F=   -0.000007  0.000007
 ITERATION  4 X=       1.32151    0.29342
NORM(F(X))=    0.000000  F=   -0.000000  0.000000
 AFTER    5 NEWTON ITERATIONS,

 THE BILINEAR CONSTRAINT HIT ZERO.
THERE ARE NOW ,  2 BUDGET SURPLUSES AT ZERO.

--------------------------------------------------
THE     6TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
          A2(*U)+                  A3
          0.3937402            1.3215135
          0.3841748            0.2934162
          0.8359918            0.0
STEPLENGTH=           0.453310  CONSTRAINT TYPE:( -1, 11)
 ITERATION  0 X=       1.50000    0.46757    0.37856
NORM(F(X))=    0.047929  F=    0.036484  0.031083  0.000000
 ITERATION  1 X=       1.50000    0.48000    0.36667
NORM(F(X))=    0.0       F=    0.0        0.0         0.0


SUPERBASIC INFO: KNU=  2  KMU=    1
NUH(I)=    0  1  2  0
MUH(I)=    1  0  0  0
```

```
  T    =     0.853333    0.480000    0.366667    0.0
LAMBDA=      1.500000    2.500000    0.500000    1.000000
 THE TIME LEFT IS NOW  1.000000 SEC.
 AFTER   1 NEWTON ITERATIONS,
 THE  18TH DUAL VARIABLE WENT TO ZERO.
 VECIN: C4AC2       VECOUT: UTIL1

-------------------------------------------------
THE     7TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
         A2(*U)+                 A3
        0.0                   0.4800000
        1.0000000             0.3666666
        0.0                   0.8533334
STEPLENGTH=            0.533334  CONSTRAINT TYPE:(  0,  9)
 ITERATION  0 X=     0.48000   0.90000   0.85333
NORM(F(X))=   0.000000  F=  -0.000000  0.0          0.000000
 AFTER    0 NEWTON ITERATIONS.

 THE BILINEAR CONSTRAINT HIT ZERO.
THERE ARE NOW .  3 BUDGET SURPLUSES AT ZERO.

-------------------------------------------------
THE     8TH ENDPNT CALL:
THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:
         A2(*U)+                 A3
        0.0                   0.4800000
        0.0                   0.9000005
        0.0                   0.8533334
        1.0000000             0.0
STEPLENGTH=            0.050000  CONSTRAINT TYPE:(  1,  5)
 ITERATION  0 X=     0.48000   0.90000   0.85333   0.05000
NORM(F(X))=   0.000000  F=  -0.000000  0.000000  0.0          0.000000
 AFTER    0 NEWTON ITERATIONS.


AFTER   8 ENDPNT CALLS, WE HAVE EQUILIBRIUM.

TOTALS:  SCALAR FUNCTION CALLS=       45
           JACOBIAN EVALUATIONS=       2

 THE TIME LEFT IS NOW  1.000000 SEC.
 UTILITY LEVELS:        1.3333334      0.8000000      6.0000006       0.9999996
 PRICES:      1.500000    2.500000    0.500000    1.000000
 WHISMAN


JH(I)                    VALUE          ROW NAMES          PI(I)           RHS
C1AC1                   1.33333337       UTIL1           1.50000000      -0.48000000
C2AC2                   0.80000001       UTIL2           2.50000000      -0.32000000
C3AC1                   6.00000057       UTIL3           0.50000000      -5.10000000
C4AC1                   0.73333267       UTIL4           1.00000000      -0.95000000
MEXP                    0.00000000       GOOD1           0.50000000       5.00000000
GOOD3                   3.39999986       GOOD2           0.50000000      11.00000000
C4AC2                   0.26666697       GOOD3           0.0              9.00000000
OBJ                    -0.00000000       OBJ            -1.00000000       0.0
```

II.5. Source Listing for BCA

The following pages are a listing of the program BCA as it is written in FORTRAN IV. The subroutines which make up LPM1 (Tomlin [1975]) are omitted.

```
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,RS,P,DD,DD1,BL1,BL2
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(300)
      DOUBLE PRECISION   E(8000)
      REAL  A(4000),C(800),CMIN,CCND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,CY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
     1A,E,CMIN,CCND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDECI,NDUAL,NIPIW,IFEAS,IFCRSH
C
      COMMON ITCH,ITCHA,IFPIWT,IFNES,KOUTB
      COMMON IA(4000),IE(8000)
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INO,INP
      COMMON/INT/ IPS(30),KDET,KCUNT,ISING
      COMMON/TOLER/ TOLFZ,TOLBD,TOLCV,STPMX,STPRD
      DIMENSION VS(10),SHR(10)
      NAMELIST/PARM1/TOLFZ,TOLBD,TOLCV,ICNTRL,IECHO,
     1STPMX,STPRD,ICECHO,IH,L,IPROD,KOUTB,ITLIME
C
C      BCA : A CODE WHICH IMPLEMENTS THE BILINEAR COMPLEMENTARITY
C   ALGORITHM FOR SOLVING ECONOMIC EQUILIBRIUM PROBLEMS.
C
C      AUTHOR:     THOMAS R. ELKEN
C                  SYSTEMS OPTIMIZATION LABORATORY
C                  OPERATIONS RESEARCH DEPARTMENT
C                  STANFORD UNIVERSITY
C
C      FOR A DESCRIPTION OF THE ALGORITHM AND DOCUMENTATION OF
C   THIS PROGRAM SEE SOL TECHNICAL REPORTS 77-26 AND 77-  .
C
      TOLBD= 1D-04
      TOLCV= 1D-05
      TOLFZ= 1D-10
      STPMX= 10.0
      STPRD= 0.5
      ITLIME= 100
      ICNTRL= 1
      IECHO= 0
      KOUTB= 0
      ICECHO= 0
      KFUN= 0
      KJAC= 0
      IPROD= 1
      READ (5,PARM1)
      WRITE (6,PARM1)
      WRITE (6,9) IECHO,IH,ICECHO
    9 FORMAT(1X,'  IECHO,IH,ICECHO',3I4)
```

```
      IF (ICNTRL) 10.40.40
   10 CALL INPUT(IACT)
      IF (IACT)  30. 20. 20
   20 IF (IECHO.EQ.0) GO TO 29
      MUB= NROW
      WRITE (6.21)
   21 FORMAT(/////.'          STRUCTURAL COLUMNS OF MATRIX.')
   22 LB= MUB+1
      MUB= LB+ 14
      K= 0
      DO 25 J= LB.MUB
          IF (J.GT.NCOL) GO TO 26
          K= K+ 1
          CALL UNPACK(J)
          DO 24 I= 1.NROW
              G1(I.K)= Y(I)
   24     CONTINUE
   25 CONTINUE
   26 WRITE (6.27)  LB.MUB
   27 FORMAT (/////.' COLUMNS '.I4.' THRUOGH '.I4)
      DO 23 I=1.NROW
      WRITE (6.28)  (G1(I.J).J=1.K)
   28 FORMAT (1X.15F8.3 )
   23 CONTINUE
      IF (MUB .LT. NCOL) GO TO 22
   29 CALL NORMAL
      CALL UNRAVL(0)
      GO TO 10
   30 STOP
C
   40 CALL INPUT(IACT)
      CALL DEBUG(1)
      CALL NORMAL
      CALL UNRAVL(0)
C
      CALL DEBUG(1)
      CALL SHIFTR(3.4)
      KEND= 0
      KENDSV= 0
      IHL= IH+ L
      M= NROW
      N= NCOL- NROW
      NM= N+ M
      L1= M- L
      MM1= M- 1
      ITSINV= 1
      IF (IPROD.EQ.0) GO TO 43
      READ (5.1030) (SHP(I).I=1.IH)
   43 ICOU=0
      ISAV= 0
      LC(1)= 1
      DO 68 K= 1.IH
          KP1= K+ 1
   45     READ(KINP.1010) ISHR. LL. KK
          IF (ISHR.EQ.1) GO TO 50
          IF (ISHR.EQ.-1) GO TO 42
          IF (LL.EQ.0) GO TO 58
          IF (KK.NE.0) GO TO 47
              ICOU= ICOU+ 1
              IC(ICOU)= LL
```

```
           C(ICOU)= B(LL)
           GO TO 45
47      DO 48 I= LL, KK
           ICOU= ICOU+ 1
           IC(ICOU)= I
           C(ICOU)= B(I)
48      CONTINUE
        GO TO 45
42     IF (KK.NE.0) GO TO 44
          ICOU= ICOU+ 1
          IC(ICOU)= LL
          READ (5,1030) C(ICOU)
          GO TO 45
44     ISAV= ICOU+ 1
       DO 46 I= LL,KK
          ICOU= ICOU+ 1
          IC(ICOU)= I
46      CONTINUE
       READ(5,1030) (C(J),J=ISAV,ICOU)
       GO TO 45
50      IF (KK.NE.0) GO TO 53
           ICOU= ICOU+ 1
           IC(ICOU)= LL
           C(ICOU)= SHR(K)*B(LL)
           GO TO 45
53      DO 55 I= LL, KK
           ICOU= ICOU+ 1
           C(ICOU)= SHR(K)*B(I)
           IC(ICOU)= I
55     CONTINUE
       GO TO 45
58     LC(KP1)= ICOU+1
68 CONTINUE
   IF (ICECHO.EQ.0) GO TO 69
   MUB= 0
   WRITE (6,61)
61 FORMAT(/////,'          STRUCTURAL COLUMNS OF MATRIX.')
62 LB= MUB+1
   MUB= LB+ IH - 1
   K= 0
   DO 65 J= LB,MUB
        IF (J.GT.NCOL) GO TO 66
        K= K+ 1
        CALL UPAKC(J)
        DO 64 I= 1,NROW
           GI(I,K)= Y(I)
64      CONTINUE
65 CONTINUE
66 WRITE (6,67)  LB,MUB
67 FORMAT (/////,' COLUMNS ',I4,' THROUGH ',I4)
   DO 74 I= 1,NROW
   WRITE (6,73) (GI(I,J),J=1,IH)
73 FORMAT (1X,10F10.4)
74 CONTINUE
69 DO 70 I= 1,M
   IR= JH(I)
   XX(IR)= X(I)
   PI(I)= YTEMP(I)
   BA(I)= X(I)
70 CONTINUE
```

```
      MPI= M+ 1
      DO 100 J= MPI,NM
      IF (KINBAS(J).NE. 0) GO TO 90
      DSUM= 0.
      LL= LA(J)
      KK= LA(J+1) - 1
      DO 80 I= LL,KK
      IR= IA(I)
      DE= A(I)
      DPROD= DE*YTEMP(IR)
      DSUM= DSUM + DPROD
   80 CONTINUE
      PI(J)= DSUM
      GO TO 100
   90 PI(J)= 0.
  100 CONTINUE
      K= 0
      DO 110 I= 1,NM
      IF (KINBAS(I) .EQ. 0) GO TO 105
      IDBAS(I)= 0
      GO TO 110
  105 K= K+ 1
      DIGMA(K)= I
      BB(K)= PI(I)
      IDBAS(I)= K
      XX(I)= 0.0
  110 CONTINUE
C       THIS IS THE BILINEAR PHASE OF THE ALGORITHM. VARIABLES XX(1).....
C   XX(M) ARE THE SLACKS. AND PI(1)....,PI(M) ARE THE USUAL PI'S.
C   XX(M+1),....,XX(M+N) ARE THE X'S, PI(M+1),....,PI(M+N) ARE THE
C   DUAL SLACKS.  MORE INITIALIZATIONS.
C
      DO 120 I=1,IH
      MUH(I)= 0
      NUH(I)= 0
      NU(I)= 0
      MU(I)= 0
  120 CONTINUE
      KNU= 0
      KMU= 0
      DD=1
  420 PD= 1
            IF (KINBAS(DD).GT.0) GO TO 425
                CALL SUPERB(1.1.DD.0.0)
                GO TO 430
  425       CALL SUPERB(1.-1.DD.0.0)
          PD= -1
  430 JS= DD
      CALL BLCONS
      ZFLAG= 0
      GO TO 460
  450 NTEMP(1)= NTEMP(1)+ JTINV
      CALL INVERT
      CALL RECALC
      ITSINV= 0
  460 KEND= KEND+ 1
      ITSINV= ITSINV + 1
      IF (KEND .GT. ITLIME) STOP
      WRITE (6,222) KEND
  222 FORMAT (/,' ----------------------------------------'
     1,/,' THE ',I4,'TH ENDPNT CALL:')
      CALL ENDPNT(JS,PD1,IS,NET)
      IF ( (KEND/3)*3 .NE. KEND ) GO TO 468
      CALL DEBUG(4)
      CALL DEBUG(1)
  468 WRITE (6,470) NET
  470 FORMAT (1X,' AFTER ',I3,' NEWTON ITERATIONS,')
      IF (PD1.EQ.1  .AND. IS.EQ.NN) GO TO 920
      IF (PD1) 300,306,310
  300 WRITE (6,305) IS
  305 FORMAT (1X,' THE ',I3,'TH DUAL VARIABLE WENT TO ZERO.')
      GO TO 480
  306 WRITE (6,307) DD
  307 FORMAT (/,1X,' THE BILINEAR CONSTRAINT HIT ZERO.',/
     1,' THERE ARE NOW ,',I3,' BUDGET SURPLUSES AT ZERO.')
      IF (DD .EQ. IH) GO TO 920
      DD= DD+ 1
      GO TO 420
  310 WRITE (6,315) IS
  315 FORMAT (1X,' THE ',I3,'TH PRIMAL VARIABLE WENT TO ZERO.')
  480 IF (PD1.EQ.-1) GO TO 520
      IF (IS .EQ. DD) GO TO 490
C
C   A PRIMAL VARIABLE WENT TO ZERO.  THAT VARIABLE MUST GO TO
C   NONBASIC. AND THE INCOMING VARIABLE IS DETERMINED BY FINDING THE
C   LARGEST ELEMENT IN THE APPROPRIATE ROW OF G1.
C
      PD= -1
  483 IROWP= KINBAS(IS)
      CALL FINDP(1,IS,JCOLP)
      INNAM1= ICNAM(JCOLP,1)
      INNAM2= ICNAM(JCOLP,2)
      IONAM1= ICNAM(IS,1)
      IONAM2= ICNAM(IS,2)
      WRITE(6,485) INNAM1,INNAM2,IONAM1,IONAM2
  485 FORMAT(1X,' VECIN: ',2A4,'  VECOUT: ',2A4)
      CALL PIVOT(JCOLP,IS)
      CALL UNPACK(JCOLP)
      CALL FTRAN(1)
      CALL WRETA
      CALL BSCNC(JCOLP,IS)
      CALL SUPERB(0,-1,JCOLP,1,JCOLP)
      CALL BLCONS
      JS= IS
      IF (ITSINV .GE. INVFRQ) GO TO 450
      GO TO 460
  490 IDS= DD
      DD= DD -1
      JS= DD
      IF (DD.EQ.0) STOP
      PD= 0
      IF (KINBAS(IDS) .NE. 0) GO TO 500
      CALL SUPERB(2,0,0,1,IDS)
      GO TO 460
C
  500 CALL SUPERB(2,0,0,-1,IDS)
      GO TO 483
C
C      A DUAL VARIABLE WENT TO ZERO.  THAT VARIABLE MUST ENTER THE
C   BASIS. AND THE LEAVING VARIABLE MUST BE DETERMINED BY FINDING
```

```
C     THE LARGEST PIVOT FLEMENT IN THE IDBAS(IS)-ROW OF G2.
C
  520 PD= 1
      IF (IS.EQ.DD) GO TO 530
  523 JCOLP= IS
      CALL FINDF(-1.IS.P)
      IRCWP= KINEAS(P)
      INNAM1= ICNAM(IS.1)
      INNAM2= ICNAM(IS.2)
      IUNAM1= ICNAM(P,1)
      IONAM2= ICNAM(F,2)
      WRITE (6.485) INNAM1.INNAM2.IONAM1.ICNAM2
      CALL PIVOT(JCOLP,P)
      CALL UNPACK(JCOLF)
      CALL FTRAN(1)
      CALL WRETA
      CALL BSCNC(JCCLP,P)
      CALL SUPERB(0,1,F,-1,F)
      CALL BLCCNS
      JS= IS
      IF (ITSINV.GE.INVFRQ) GO TO 450
      GO TC 460
  530 PD= 0
      IDS= DD
      DD= DD- 1
      JS= DD
      IF (DD.EG.0) STOF
      IF (ICBAS(IDS) .NE. 0) GC TC 540
      CALL SUPERB(2.0.C.-1.IDS)
      GO TO 460
C
  540 CALL SUPERB(2,0.C.1.IDS)
      GO TO 523
C
C
  920 WRITE (6,930) KEND
  930 FORMAT(//,' AFTER'.I4.' ENDENT CALLS. WE HAVE EQUILIBRIUM.')
      KJAC= KJAC/ (IH*IH)
      WRITE (6,320) KFUN.KJAC
  320 FORMAT(/,' TOTALS:  SCALAR FUNCTION CALLS=',I8./.11X.'JACOBIAN EVA
     1LUATIONS=',I8./)
      CALL DEBUG(1)
      DO 935 I= 1.IH
  935 VS(I)= -B(I)+ XX(I)
      WRITE (6,940) (VS(I).I=1.IH)
      WRITE (6,945) (PI(I),I= 1. IH )
  940 FORMAT (1X,' UTILITY LEVELS: '.6F15.7)
  945 FORMAT (1X,' PRICES: '.9F12.6)
      DO 950 I= 1.NROW
         IV= JH(I)
         X(I)= XX(IV)
         Y(I)= PI(I)
  950 CONTINUE
      CALL UNRAVL(0)
      STOP
 1010 FORMAT (15I4)
 1030 FORMAT (10F7.4 )
      END
      SUBROUTINE BLCONS
C
```

```
C      WE ARE GOING TO CALCULATE THE DD ROWS OF COEFFICIENTS
C    FOR THE BILINEAR EQUATIONS AND THE BILINEAR INEQUALITY.
C      THE BASIC MATHEMATICAL STRUCTURE IS THE FOLLOWING:
C
C         BF1 + D1*PI(MU)- DIAG(PI(MU))*(F1*X(NU) + E1) = 0
C         BF2 + D2*PI(MU)- DIAG(X(NU))*(F2*PI(MU) + E2) = 0
C
       IMPLICIT REAL*8 (A-H,O-Z)
       INTEGER PD,PD1,PD2,SS,RR,ZFLAG,RS,P,DD,DD1,BL1,BL2
       INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
       INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
       DOUBLE PRECISION  E(8000)
       REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
       COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
      1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
      2NTEMP(20),KINF,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
      3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
      4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
      5LA(1302),LE(2002),PUN(8),
      6IPUNC,NDEGI,NDUAL,NIPIW,IFEAS,IFCRSH
       COMMON ITCH,ITCHA,IFPIWT,IFNEG,KCUTR
       COMMON IA(4000),IE(8000)
       COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
       COMMON/LP1/PI(1302),XX(1302)
       COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
       COMMON/LNCCNS/G1(350,10),G2(400,10),BA(350),BB(400)
       COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
       COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
       COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
       COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
       K1= 0
       K2= 0
       BL1= 0
       BL2= 0
       INQ= MUH(DD)
       INP= NUH(DD)
       IF (KMU.EQ.0) GO TO 100
       DO 80 K= 1,KMU
       K1= MU(K)
       IF (K1.GT.IH) GO TO 80
       IDD= KINBAS(K1)
       BL1= BL1+ 1
       IF (KNU.EQ.0) GO TO 45
       DO 40 I= 1,KNU
    40 F1(K,I)= G1(IDD,I)
    45 E1(K)= BA(IDD)
       CALL UPAKC(K1)
       DO 60 I= 1,KMU
       DSUM  = 0.0
       IF (MU(I) .LE. M)    DSUM= Y(MU(I))
       DO 50 J= 1,M
          IDJ= IDBAS(J)
          IF (IDJ .EQ. 0) GO TO 50
          DSUM= DSUM+ Y(J)*G2(IDJ,I)
    50 CONTINUE
       D1(K,I)= DSUM
    60 CONTINUE
       BF1(K)= 0.
       DO 70 J= 1,M
      IF(IDBAS(J).EQ.0) GO TO 70
      BF1(K)= BF1(K)+     Y(J)*BE(IDBAS(J))
   70 CONTINUE
      IF (K1.NE.DD) GO TO 80
      INEG= 1
      BL1= BL1-1
   80 CONTINUE
C
C     NOW THE SECOND TYPE OF EQUATION IS CALCULATED BECAUSE
C   PI(K) IS BASIC.
C
  100 IF (KNU.EQ.0) RETURN
      DO 145 K= 1,KNU
      K2= NU(K)
      IF(K2.GT.IH) GO TO 145
      IDD= IDBAS(K2)
      BL2= BL2+ 1
      IF (KMU.EQ.0) GO TO 123
      DO 120 I= 1,KMU
  120 F2(K,I)= G2(IDD,I)
  123 E2(K)= BB(IDD)
      CALL UPAKC(K2)
      IF (KNU.EQ. 0) GO TO 132
      DO 130 I= 1,KMU
      D2(K,I)= 0.0
      IF (MU(I) .LE. M)    D2(K,I)= Y(MU(I))
      DO 125 J= 1,M
      IF( IDBAS(J).EQ.0) GO TO 125
      D2(K,I)= D2(K,I)+ Y(J)*G2(IDBAS(J),I)
  125 CONTINUE
  130 CONTINUE
  132 BF2(K)= 0.
      DO 140 J= 1,M
      IF (IDBAS(J).EQ.0) GO TO 140
      BF2(K)= BF2(K)+ Y(J)*BB(IDEAS(J))
  140 CONTINUE
      IF (K2. NE. DD) GO TO 145
      INEG= 2
      BL2= BL2- 1
  145 CONTINUE
      RETURN
      END
      SUBROUTINE FINDP(PD1,IS,P)
C
C     THIS SUBROUTINE CHOOSES THE VARIABLE TO BECOME IMPLICITLY
C   BASIC AS THE ONE WITH THE PIVOT ELEMENT LARGEST IN
C   ABSOLUTE VALUE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,SS,FF,ZFLAG,RS,F,DD,DD1,BL1,BL2
      INTEGER*2 JH(350),SIGMA(952),KINBAS(1302),IDBAS(1302)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/ JH,SIGMA,KINBAS,IDBAS
      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INO,INP
      COMMON/TOLER/ TOLEZ,TOLBD,TOLCV,STEMX,STPRD
      P= 0
      IF (PD1.EQ.-1) GO TO 30
      IDD= KINBAS(IS)
      DO 20 I= 1,KNU
      IF (P.NE.0) GO TO 10
      BIG=  DABS(G1(IDD,I))
      P= NU(I)
      GO TO 20
   10 COMP= DABS(G1(IDD,I))
      IF (COMP.LE.BIG) GO TO 20
      BIG= COMP
      P= NU(I)
   20 CONTINUE
      IF (BIG.LT.TOLFZ) P= 0
      RETURN
   30 IDD= IDBAS(IS)
      DO 40 J= 1,KMU
      IF (P.NE.0) GO TO 35
      BIG= DABS(G2(IDD,J))
      P= MU(J)
   35 COMP= DABS(G2(IDD,J))
      IF (COMP.LE.BIG) GO TO 40
      BIG= COMP
      P= MU(J)
   40 CONTINUE
      IF (BIG.LT.TOLFZ) P= 0
      RETURN
      END
      SUBROUTINE PIVOT (SS,RR)
C
C       THIS ROUTINE PERFORMS A PIVOT ON THE PRIMAL SUPER-
C     BASIC COLUMNS IF PD.EQ.1. DUAL IF PD.EQ.-1. THE PIVOT
C     BRINGS COLUMN SS INTO THE BASIS AND COLUMN RR OUT OF THE
C     BASIS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,RS,P,DD,DD1,BL1,BL2
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),ZB(350),
     1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFFZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDECI,NDUAL,NIPIW,IFBAS,IFCRSH
      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
      COMMON IA(4000), IE(8000)
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INO,INP
      EPS= 16.**(-13)
      RB= KINBAS(RR)
      IF (SS .GT. IH) GO TO 5
      IF (NUH(SS).NE.0) GO TO 20
    5 CALL UNPACK(SS)
      CALL FTRAN(1)
      ZB (RB)= -1./Y(RB)
      DO 10 I= 1,M
      IF (I.NE.RB) ZB(I)= Y(I)*ZB(RB)
   10 CONTINUE
      GO TO 30
   20 JS= NUF(SS)
      ZB(RB)= 1./G1(RB,JS)
      DO 25 I= 1,M
      IF (I.NE.RB) ZB(I)= -G1(I,JS)*ZB(RB)
   25 CONTINUE
   30 IF (KNU.EQ.0) GO TO 55
      DO 50 J= 1,KNU
      IF (NU(J).EQ.SS) GO TO 50
      V= G1(RB,J)
      G1(RB,J)= 0.
      DO 40 I= 1,M
   40 G1(I,J)= G1(I,J)+ V*ZB(I)
      G1(RB,J)= -G1(RB,J)
   50 CONTINUE
   55 V= BA(RB)
      BA(RB)= 0.
      DO 60 I= 1,M
   60 BA(I)= BA(I)+ V*ZB(I)
      BA(RB)= -BA(RB)
C
C        WE ARE NOW PIVOTING ON THE DUAL SYSTEM.  IF THIS
C     IS NOT AN IMPLICIT BASIC-TYPE PIVOT, WE CALL SUPERB
C     TO CALCULATE THE PIVOT COLUMN.
C
      IFL= 0
      RB= IDBAS(SS)
      IF (RB .GT. IH) GO TO 70
      IF (MUF(RB).NE.0) GO TO 80
   70 IFL= 1
      CALL SUPERB(1,-1,RB,0,0)
      JS= KMU
      GO TO 82
   80 JS= MUF(RB)
   82 ZB(RB)= 1./G2(RB,JS)
      DO 85 I= 1,N
      IF (I.NE.RB) ZB(I)= -G2(I,JS)*ZB(RB)
   85 CONTINUE
      IF (KMU.EQ.0) GO TO 115
      DO 110 J= 1,KMU
      IF (MU(J).EQ.RB) GO TO 110
      V= G2(RB,J)
      G2(RB,J)= 0.
      DO 100 I= 1,N
  100 G2(I,J)= G2(I,J)+ V*ZB(I)
      G2(RB,J)= -G2(RB,J)
  110 CONTINUE
  115 V= BB(RB)
      BB(RB)= 0.0
      DO 120 I= 1,N
  120 BB(I)= BB(I)+ V*ZB(I)
      BB(RB)= -BB(RB)
      IF (IFL.EQ.0) RETURN
      CALL SUPERB (2,0,0,-1,RB)
      RETURN
      END
C
C
```

```
      SUBROUTINE UPAKC(II)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IF,PUN,LC(20),IC(800)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DF,B(350),X(350),Y(350),YTEMP(350),
     1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDEGI,NDUAL,NIPIW,IFBAS,IFCRSH
      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTP
      COMMON IA(4000),IF(8000)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
C
      DO 100 I= 1,NROW
      Y(I)= 0.
  100 CONTINUE
C
      LL= LC(II)
      KK= LC(II+1) - 1
      DO 200 I= LL,KK
      IR= IC(I)
      Y(IR)= C(I)
  200 CONTINUE
C
      RETURN
      END
C
C        THE FOLLOWING ROUTINE MAKES THE CHANGES IN THE INDEX
C   SETS NECESSARY EVERY TIME A BASIS CHANGE IS MADE.
C
      SUBROUTINE BSCNG(S,R)
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      INTEGER  S,SSAV,R,RSAV
      RSAV= KINBAS(R)
      JH(RSAV)= S
      KINBAS(R)= 0
      KINBAS(S)= RSAV
      SSAV= IDBAS(S)
      DIGMA(SSAV)= R
      IDBAS(S)= 0
      IDBAS(R)= SSAV
      RETURN
      END
C       THIS SUBROUTINE CAN DO THREE THINGS DETERMINED BY THE
C   PARAMETER 'KEY'.  IT CAN ADD A COLUMN, REMOVE A COLUMN, OR
C   BOTH ADD AND REMOVE COLUMNS FROM THE PRIMAL OR DUAL SUPER-
C   BASIC COLUMNS, DEPENDING ON WHETHER KEY IS 1, 2, OR 0, RESP-
C   ECTIVELY.  PD1 IS 1 OR -1 DEPENDING ON WHETHER A PRIMAL OR
C   DUAL COLUMN IS BEING ADDED.  PD2 IS A SIMILAR FLAG FOR THE
C   COLUMN BEING REMOVED.  IS AND JS INDICATE THE PARTICULAR
C   COLUMN TO BE ADDED OR REMOVED FROM THE GROUP OF COLUMNS
C   SPECIFIED BY PD1 AND PD2.
C
      SUBROUTINE SUPERB(KEY,PD1,IS,PD2,JS)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,SS,FR,ZFLAG,RS,P,CD,DD1,BL1,BL2
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,CCND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
     1A,E,CMIN,CCND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDEGI,NDUAL,NIPIW,IFBAS,IFCRSH
      COMMON ITCF,ITCHA,IFPIWT,IFNEG,KOUTB
      COMMON IA(4000),IE(8000)
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
      EPS= 16.**(-13)
      IF (KEY.EQ.1) GO TO 50
      IF (PD2.GT.0) GO TO 20
      KM= KMU
      KMU= KMU- 1
      IF (JS .GT. IH) GO TO 4
      MH= MUH(JS)
      MUH(JS)= 0
      IF (MH.EQ.KM) GO TO 6
      DO 5 I= MH,KMU
      IM= MU(I+1)
      MU(I)= IM
       MUH(IM)= MUH(IM)- 1
      DO 5 J= 1,N
      G2(J,I)= G2(J,I+1)
    5 CONTINUE
    6 MU(KM)= 0
    4 DO 7 J= 1,N
    7 G2(J,KM)= 0.0
      IF (KEY.EQ.2) RETURN
      GO TO 50
   20 KN= KNU
      KNU= KNU- 1
      IF (JS .GT. IH) GO TO 28
      NH= NUH(JS)
      NUH(JS)= 0
      IF (NH.EQ.KN) GO TO 26
      DO 25 I= NH,KNU
      JN= NU(I+1)
      NU(I)= JN
      NUH(JN)= NUH(JN)- 1
      DO 25 J= 1,M
      G1(J,I)= G1(J,I+1)
   25 CONTINUE
   26 NU(KN)= 0
   28 DO 30 I= 1,M
   30 G1(I,KN)= 0.0
      IF (KEY.EQ.2) RETURN
C
```

```
C        NOW WE ADD THE SUPERBASIC VARIABLE SPECIFIED BY PD1.IS.
C
      50 IF (PD1.LT.0) GO TO 70
         KN= KNU
         KNU= KNU+ 1
         IF (IS.GT.IH) GO TO 55
         NU(KNU)= IS
         NUH(IS)= KNU
      55 CALL UNPACK(IS)
         CALL FTRAN(1)
         DO 60 I= 1,M
      60 G1(I,KNU)= -Y(I)
         RETURN
C
C          NOW WE ARE BRINGING A DUAL SUPERBASIC COLUMN IN.
C
      70 KMU= KMU+ 1
         IF (IS.GT.IH) GO TO 73
         MU(KMU)= IS
         MUH(IS)= KMU
      73 IF (IS.LE.M) GO TO 110
C
C         THE ENTERING VARIABLE IS A ZETA.  THE ENTERING COLUMN
C     IS A ROW OF INV(B) AND A ROW OF INV(B)*D.
C
         IDD= KINBAS(IS)
         DO 75 I= 1,M
      75 Y(I)= 0.
         Y(IDD)= 1.
         CALL BTRAN
         CALL SHIFTR(3,2)
         DO 80 I= 1,M
         IF (IDBAS(I).EQ.0) GO TO 80
         G2(IDBAS(I),KMU)= X(I)
      80 CONTINUE
         DO 100 J= MP1,NM
         IF (IDBAS(J).EQ.0) GO TO 100
         CALL UNPACK(J)
         DOT= 0.0
         DO 90 I= 1,M
         IF (IDBAS(I).EQ.0) GO TO 90
         DOT= DOT+ Y(I)*X(I)
      90 CONTINUE
         G2(IDBAS(J),KMU)= DOT
     100 CONTINUE
         RETURN
C
C          THE ENTERING VARIABLE IS A PI.  THE ENTERING COLUMN
C     IS PARTLY C1*INV(B) AND PARTLY C2+ C1*INV(B)*D.
C     -C1*INV(B) IS PART OF THE BASIS INVARSE CORRESPONDING TO
C     THE X-BASIC COLUMNS AND THE T-BASIC ROWS.
C
     110 IDD= KINBAS(IS)
         DO 130 I= 1,M
     130 Y(I)= 0.
         Y(IDD)= 1.
         CALL BTRAN
         CALL SHIFTR(3,2)
         DO 140 I= 1,M
         IF (IDBAS(I).EQ.0) GO TO 140
      G2(IDBAS(I),KMU)= X(I)
  140 CONTINUE
      DO 160 J= MP1,NM
      IF (IDBAS(J).EQ.0) GO TO 160
      CALL UNPACK(J)
      DOT= 0.0
      DO 150 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 150
      DOT= DOT+ Y(I)*X(I)
  150 CONTINUE
      G2(IDBAS(J),KMU)= Y(IS)+ DOT
  160 CONTINUE
      RETURN
      END
      SUBROUTINE RECALC
C
C        THIS SUBROUTINE RECALCULATES THE SUPERBASIC COLUMNS
C   USING THE CURRENT BASIS IN OR FORM.  IT CAN ALSO ADD
C   A VARIABLE AND COLUMN TO THE SUPERBASICS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,RS,P,PO,DD1,BL1,BL2
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(80)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
     1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDECI,NDUAL,NIPIW,IFEAS,IFCRSH
      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
      COMMON IA(4000),IE(8000)
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/UNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUF(10),MUF(10),NU(10),MU(10)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PO,MPD,INEQ,KFUN,KJAC
      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
      EPS= 16.**(-13)
      IF (KNU.EQ.0) GO TO 26
      DO 28 J= 1,KNU
      IDD= NU(J)
      CALL UNPACK(IDD)
      CALL FTRAN(1)
      DO 25 I= 1,M
   25 G1(I,J)= - Y(I)
   28 CONTINUE
   26 CALL SHIFTR(1,3)
      CALL FTRAN(1)
      DO 22 I= 1,M
   22 BA(I)= Y(I)
   30 IF (KMU.EQ.0) GO TO 140
      DO 130 J= 1,KMU
      IDD= MU(J)
      IF (IDD.LE.M) GO TO 70
C
```

```
C     ZETA- VARIABLE IS ENTERING
C
      IB= KINBAS(IDD)
      DO 35 I= 1,M
   35 Y(I)= 0.
      Y(IB)= 1.
      CALL BTRAN
      CALL SHIFTR(3.2)
      DO 40 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 40
      G2(IDBAS(I),J)= X(I)
   40 CONTINUE
      DO 60 K= MP1,NM
      IF (IDBAS(K).EQ.0) GO TO 60
      CALL UNPACK(K)
      DOT= 0.0
      DO 50 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 50
      DOT= DOT+ Y(I) * X(I)
   50 CONTINUE
      G2(IDBAS(K),J)= DOT
   60 CONTINUE
      GO TO 130
C
C     PI- VARIABLE IS ENTERING
C
   70 IB= KINBAS(IDD)
      DO 90 I= 1,M
   90 Y(I)= 0.
      Y(IB)= 1.
      CALL BTRAN
      CALL SHIFTR(3.2)
      DO 100 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 100
      G2(IDBAS(I),J)= X(I)
  100 CONTINUE
      DO 120 K= MP1,NM
      IF (IDBAS(K).EQ.0) GO TO 120
      CALL UNPACK(K)
      DOT= 0.0
      DO 110 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 110
      DOT= DOT+ Y(I) *X(I)
  110 CONTINUE
      G2(IDBAS(K),J)= Y(IDD)+ DOT
  120 CONTINUE
  130 CONTINUE
  140 DO 150 I= 1,M
  150 Y(I)= 0.0
      IF (KINBAS(NM).NE.0)  Y(KINBAS(NM))= 1.
      CALL BTRAN
      CALL SHIFTR(3.2)
      DO 160 I= 1,M
  160 IF (IDBAS(I).NE.0)  BB(IDBAS(I))= X(I)
      DO 170 J= MP1,NM
      IF (IDBAS(J).EQ.0) GO TO 170
      CALL UNPACK(J)
      DOT= 0.0
      DO 165 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 165
      DOT= DOT+ X(I)*Y(I)
  165 CONTINUE
      BB(IDBAS(J))= DOT
  170 CONTINUE
      RETURN
      END
      SUBROUTINE ENDPNT(JS,PD1,IS,NET)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MIN,MIN2
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(80))
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,P,DD,DD1,BL1,BL2
      REAL C(800)
      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10)
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/LNCONS/G1(350,10),G2(400,10),EA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
      COMMON/INT/ IPS(30),KDET,KOUNT,ISING
      COMMON/TOLER/ TOLFZ,TOLBD,TOLCV,STPMX,STPRD
      DIMENSION U(3),V1(10),V2(10),F(10),DOT(4),RHS(10),UL(10,10)
C
C       THIS PROGRAM WILL FIND THE ENDPOINT OF A CURVE
C     DEFINED BY A SYSTEM OF DD-1 BILINEAR EQUATIONS IN DD
C     UNKNOWNS.  ONE ENDPOINT IS KNOWN, AND THE OTHER END-
C     POINT IS DETERMINED BY THE FIRST INTERSECTION OF THE
C     CURVE WITH ONE OF NM+1 INEQUALITY CONSTRAINTS.
C        THE SYSTEM IS
C
C     BF1+ D2*X(MU)- DIAG(X(MU))*(F1*X(NU)+ E1)= 0
C     BF2+ D2*X(MU)- DIAG(X(MU))*(F2*X(MU)+ F2)= 0
C
C       WHERE ONE OF THE ABOVE IS AN INEQUALITY IF ZFLAG.NE.1/
C     THE INEQUALITY IS THE LAST IN THE GROUP DEFINED BY INEQ.
C     THE LINEAR INEQUALITIES ARE DEFINED BELOW.
C
      NET= 0
      FRAC=1.
C        MFLAG=0 WHEN WE ARE SOLVING FOR THE INITIAL 3 POINTS
C        MFLAG=1 WHEN ONE OF THE G-CONSTRAINTS IS BINDING.
C        MFLAG=2 WHEN ONE OF THE VARIABLES GOES TO ZERO.
C        MFLAG=3 WHEN WE USE THE NORMAL HYPERPLANE TO DEFINE THE
C                SUBPROELEM SHORT OF THE BOUNDARY OF THE CELL.
      ITER= 0
      ISGCT= 0
      MFLAG=0
      L=1
      KK= 0
      DD1= DD- 1
      ID= INQ
      KMU1= KMU+ 1
      IF (INEQ .EQ. 2) ID= INP+ KMU
      IF (PD.EQ.0) ID= DD
      ICX= ID
      IF (KMU.EQ.0) GO TO 615
      DO 610 I= 1,KMU
  610 X(I)= FI(MU(I))
  615 IF (KNU.EQ.0) GO TO 625
      DO 620 I= 1,KNU
      KM= KMU+ 1
  620 X(KM)= XX(NU(I))
  625 ALPHA= -1.
    5 ITER= ITER+ 1
      IF (DD.GT.1) GO TO 6
            KDET= 1
            GO TO 41
    6 IF (KNU.EQ.DD .OR. KMU.EQ.DD) GO TO 630
      CALL DERIV
    7 ISGCT= ISGCT+ 1
      IF (ISGCT.GT.DD) GO TO 300
      DO 10 I= 1,DD1
          RHS(I)= -H(I,ICX)
   10 CONTINUE
      IF (ICX.EQ.DD) GO TO 25
      DO 24 J= ICX,DD1
           JP1= J+ 1
           DO 23 I= 1,DD1
   23               H(I,J)= H(I,JP1)
   24 CONTINUE
   25 IF (DD1.GT.1) GO TO 35
      IF (DABS(H(1,1)) .GE. TOL(V) GO TO 26
          ISING= 1
          GO TO 37
   26      V2(1)= RHS(1)/H(1,1)
           IF (H(1,1).GE.0) GO TO 33
                KDET= -1
                GO TO 40
   33      KDET= 1
           GO TO 40
   35 CALL DECOMP(DD1,H,UL)
   37 IF (ISING.NE.1) GO TO 38
          IF (ICX .EQ. 1) GO TO 36
          ICX= ICX- 1
          GO TO 7
   36     ICX= DD
          GO TO 7
   38 CALL SOLVE(DD1,UL,RHS,V2)
   40 DO 39 I= 1,DD1
          IM= DD- I
          IF (IM.LT.ICX) GO TO 41
              IP1= IM+ 1
              V2(IP1)= V2(IM)
   39 CONTINUE
      GO TO 41
  630 KDET= 1
      DO 634 I= 1,DD
           V2(I)= 0.0
  634 CONTINUE
   41 V2(ICX)= 1.0
      IF (ITER.GT.1) GO TO 45
      SUM= 0.0
      IF (PD.EQ.0) GO TO 120
      IF (PD.EQ.-1) GO TO 110
       JB= KINBAS(JS)
      IF (JB.NE.0) GO TO 104
             IF (V2(ID).GT.0) GO TO 135
```

```
              GO TO 140
   104 DO 105 I= 1,KNU
            IK= KMU+ 1
            SUM= SUM+ V2(IK)*G1(JB,I)
   105 CONTINUE
        IF (SUM.LT.0.0) GO TO 140
        GO TO 135
   110 JB= ICBAS(JS)
        IF (JB.NE.0) GO TO 112
              IF (V2(ID).GT.0) GO TO 135
              GO TO 140
   112 DO 115 I= 1,KMU
            SUM= SUM+ V2(I)*G2(JB,I)
   115 CONTINUE
        IF (SUM.LT. 0.0) GO TO 140
        GO TO 135
C
C     CALCULATE THE GRADIENT OF THE SURFACE, F(DD)= 0.0.
C
   120 IF (INEQ.EQ.2) GO TO 126
C          F(DD)= BF1(INQ)+ D1(INQ,.)*PI(MU)- PI(INQ)*
C                           (F1(INQ,.)*XX(NU)+ E1(INQ)).
        INQ= MUH(DD)
        DO 122 I= 1,KMU
   122      V1(I)= D1(INQ,I)
        DO 123 J= 1,KNU
            JK= J+ KMU
   123      SUM= SUM+ F1(INQ,J)*X(JK)
        V1(INQ)= V1(INQ)- SUM- E1(INQ)
        PID= X(INQ)
        DO 124 J= 1,KNU
            JK= J+ KMU
   124      V1(JK)= - PID*F1(INQ,J)
        SUM= 0.0
        DO 125 I= 1,DD
   125      SUM= SUM+ V1(I)*V2(I)
        IF (SUM.LT.0) GO TO 140
        GO TO 135
C
C    F(DD)= BF2(INQ)+ D2(KNU,.)*PI(MU)- X(IPQ)*(F2(INP,.)*PI(MU)+E2)
C
   126 IB= INP+ KMU
        DO 127 J= 1,KMU
   127      SUM= SUM+ F2(INP,J)*X(J)
        PID= X(IB)
        DO 128 J= 1,KMU
            V1(J)= D2(INP,J)- PID*F2(INP,J)
   128 CONTINUE
        DO 129 J= KMU1,DD
   129      V1(J)= 0.0
        V1(IB)= -SUM-E2(INP)
        SUM= 0.0
        DO 130 I= 1,DD
   130      SUM= SUM+ V1(I)*V2(I)
        IF (SUM.LT.0) GO TO 140
   135 INIT= KDET
        GO TO 45
   140 INIT= -KDET
    45 CALL NORM(V2,S1,DD)
        S1= (KDET*INIT)/S1
```

```
      DO 47 I= 1,DD
            ACC(2,I)= V2(I)*S1
            ACC(3,I)= X(I)
   47 CONTINUE
      WRITE (6,850)
      WRITE (6,855)   ((ACC(I,J),I=2,3),J=1,DD)
C
C         FIND THE FIRST BOUNDARY THAT THE TANGENT, Q(ALPHA)
C     HITS.  WE ARE TRYING TO FIND
C     THE SMALLEST ALPHA SUCH THAT
C           G(I,.)*Q(ALPHA) + GE(I) = 0
C
      IFLAG= 0
      IF (KNU.EQ.0) GO TO 249
      DO 250 I= 1,M
      IV= JH(I)
      IF (IV.EQ.M .OR. IV.LE.IH) GO TO 250
         DO 240 LL= 2,3
            U(LL)= 0.0
            DO 230 K= 1,KNU
               KJ= K+ KNU
               U(LL)= U(LL)+ G1(I,K)*ACC(LL,KJ)
  230       CONTINUE
  240    CONTINUE
  241    U(3)= U(3)+ EA(I)
         IF (DABS(U(2)).LT.TOLFZ) GO TO 250
         ALPHA= -U(3)/U(2)
         IF (ALPHA .LT. -TOLFZ) GO TO 250
         IF (ALPHA .GT. TOLBD) GO TO 244
            TOD= 0.0
            DO 242 K= 1,KNU
               KN= K+ KNU
  242          TOD= TOD+ G1(I,K)*ACC(2,KM)
            IF (TOD .GT. -TOLFZ) GO TO 250
  244    IF (IFLAG.EQ.1) GO TO 243
         IFLAG = 1
         JJ= I
         MIN= ALPHA
         MPD= 1
         GO TO 250
  243    IF (ALPHA.GE.MIN)  GO TO 250
         JJ= I
         MIN= ALPHA
         MPD= 1
  250 CONTINUE
  249 IF (KMU.EQ.0) GO TO 640
      DO 259 I= 1,N
         IF (DIGMA(I) .EQ. M) GO TO 259
         DO 256 LL= 2,3
            U(LL)= 0.0
            DO 254 K= 1,KNU
               U(LL)= U(LL)+ G2(I,K)*ACC(LL,K)
  254       CONTINUE
  256    CONTINUE
  257    U(3)= U(3)+ EB(I)
         IF (DABS(U(2)).LT.TOLFZ) GO TO 259
         ALPHA= -U(3)/U(2)
         IF (ALPHA .LT. -TOLFZ) GO TO 259
         IF (ALPHA.GT. TOLBD) GO TO 258
            TOD= 0.0
```

```
                  DC 252 K= 1.KMU
                       TOD= TOD+ ACC(2.K)*G2(I.K)
252               CONTINUE
                  IF (TOD .GT. -TOLFZ) GO TO 259
258          IF (IFLAG.EQ.1) GO TO 253
             IFLAG = 1
             JJ= I
             MIN= ALPHA
             MPD= -1
             GO TO 259
253          IF (ALPHA.GE.MIN)  GO TO 259
             JJ= I
             MIN= ALPHA
             MPD= -1
259   CONTINUE
640   MFLAG= 1
      IMAG= 0
      DO 588 I= 1,4
588          DOT(I)= 0.0
590   IF (INEQ.EQ.2) GO TO 600
      DO 592 I= 1,DD
              IF (I.GT.KMU) GO TO 591
              DOT(1)= DOT(1)+ C1(INQ,I)*ACC(3,I)
              DOT(2)= DOT(2)+ D1(INQ,I)*ACC(2,I)
              GO TO 592
591           IK= I- KMU
              DOT(3)= DOT(3)+ F1(INQ,IK)*ACC(3,I)
              DOT(4)= DOT(4)+ F1(INQ,IK)*ACC(2,I)
592   CONTINUE
      U(1)= -ACC(2,INQ)*DOT(4)
      U(2)= -(ACC(2,INQ)*E1(INQ)+ ACC(3,INQ)*DOT(4))+ DOT(2)
      U(3)= BF1(INQ)+ DOT(1)- ACC(3,INQ)*(DOT(3)+ F1(INQ))
      CALL QUADS(U,IMAG,ALPHA,BETA)
      IF (ALPHA.LT.TOLCV) ALPHA= BETA
      GO TO 605
600   IB= INP+ KMU
      IF (KMU.EQ.0) GO TO 603
      DO 602 I= 1.KMU
              DOT(1)= DOT(1)+ C2(INP,I)*ACC(3,I)
              DOT(2)= DOT(2)+ D2(INP,I)*ACC(2,I)
              DOT(3)= DOT(3)+ F2(INP,I)*ACC(3,I)
              DOT(4)= DOT(4)+ F2(INP,I)*ACC(2,I)
602   CONTINUE
603   U(1)= -ACC(2,IB)*DOT(4)
      U(2)= -(ACC(2,IB)*E2(INP)+ ACC(3,INP)*DOT(4))+ DOT(2)
      U(3)= BF2(INP)+ DOT(1)- ACC(3,INP)*(DOT(3)+ F2(INP))
      CALL QUADS(U,IMAG,ALPHA,BETA)
      IF (ALPHA.LT.TOLCV) ALPHA= BETA
605   IF (IMAG.EQ.1) GO TO 260
      IF (BETA .LT. -TOLCV) GO TO 260
      IF (ALPHA .LT. -TOLCV) ALPHA= BETA
      IF (ALPHA .GT. MIN) GO TO 260
      JJ= M+ 1
      MIN= ALPHA
      MPD= 0
C
C     NOW CHECK TO SEE IF ONE OF THE VARIABLES GOES TO ZERO
C     BEFORE ANY OF THE CONSTRAINTS BECOME TIGHT.
C
260   MFLAG= 1
```

```
      DO 280 I=1,DD
      DO 265 LL=2,3
      U(LL)= ACC(LL,I)
  265 CONTINUE
      IF (DABS(U(2)) .LT. TOLFZ) GO TO 280
      ALPHA= -U(3)/U(2)
      IF (ALPHA .GT. MIN .OR. ALPHA .LT. TOLFZ) GO TO 280
          JJ = I
          MFLAG= 2
          MIN= ALPHA
  280 CONTINUE
      WRITE (6,635) MIN, MPD, JJ
  635 FORMAT(' STEPLENGTH= ',F15.6,'  CONSTRAINT TYPE:(',I3,',',I3,')')
      IF (MIN .LT. STEMX) GO TO 255
          MIN= MIN* STPRD
          MFLAG= 3
          WRITE (6,635) MIN, MPD, JJ
  255 BT= 0.0
      DO 290 I= 1,DD
          X(I)= MIN * ACC(2,I)   +  ACC(3,I)
          BT= BT + X(I)*ACC(2,I)
  290 CONTINUE
C
C         NEWTON'S METHOD FOR THE PROBLEM OF FINDING THE INTERSECTION
C     OF THE CURVE WITH THE CONSTRAINT DEFINED BY MPD AND JJ.
C
  295 KK= KK+ 1
      IF (KK.GE.5) STOP
      DO 100 K= 1,15
          KM1= K- 1
          IRWDIM= DD1
          IF(MFLAG.GE.1) IRWDIM= DD
          CALL FTN(F,X)
          CALL NORM(F,S1,IRWDIM)
          WRITE (6,54) KM1,(X(I),I=1,DD)
   54     FORMAT ('  ITERATION',I3,' X= ',9F10.5)
         WRITE (6,936) S1, (F(I),I=1,IRWDIM)
          IF (S1.LT.TOLCV) GO TO 800
          CALL DERIV
          NET= NET+ 1
          IF (DD.GT.1) GO TO 55
              IF (DABS(H(1,1)) .LT. TOLCV) GO TO 58
              Z(1)= F(1)/H(1,1)
              GO TO 65
   55     CALL DECOMP(IRWDIM,H,UL)
          IF (ISING.EQ.0) GO TO 60
   58     WRITE (6,940)
          STOP
   60     CALL SOLVE(IRWDIM,UL,F,Z)
   65     ALPH= 1.0
          CALL DSENT(ALPH,S1)
          IF (ALPH.GT. TOLFZ) GO TO 70
          WRITE (6,945)
          GO TO 800
   70     DO 95 I= 1,DD
          X(I)= X(I)- ALPH * Z(I)
   95     CONTINUE
  100 CONTINUE
      MFLAG= 4
      MIN= MIN/2.
      IF (MIN.GT.TOLBD) GO TO 255
          GO TO 300
  800 IF (MFLAG.EQ.3) GO TO 5
      CALL CONCHK(GMIN,KGMIN,MP2)
       IF (KMU.EQ.0) GO TO 146
      DO 145 I= 1,KMU
          KM= NU(I)
          PI(KM)= X(I)
          IF (X(I).GE.GMIN) GO TO 145
              GMIN= X(I)
              KGMIN= KM
              MP2= -1
  145 CONTINUE
  146 IF (KNU.EQ.0) GO TO 148
      DO 147 I= KMU1,DD
          IK= I- KMU
          KM= NU(IK)
          XX(KM)= X(I)
          IF (X(I).GE.GMIN) GO TO 147
             GMIN= X(I)
             KGMIN= KM
             MP2= 1
  147 CONTINUE
  148 IF (GMIN .LT. -TOLBD) GO TO 170
      IF (MPD .NE. 0) GO TO 153
      PD1= 0
      RETURN
  153 IF (MFLAG.EQ.2) GO TO 150
      IF (MPD.EQ. -1) GO TO 151
      IS= JB(JJ)
      PD1= 1
      RETURN
  151 IS= SIGMA(JJ)
      PD1= -1
      RETURN
  150 IF (JJ.GT.KMU) GO TO 152
      IS= MU(JJ)
      PD1= -1
      RETURN
  152 JJ= JJ- KMU
      IS= NU(JJ)
      PD1= 1
      RETURN
C
C         IF THE CURRENT VALUE OF X VIOLATES THE CONSTRAINT DEFINED
C    BY KGMIN AND MP2, FIND A GOOD STARTING POINT FOR NEWTON'S
C    METHOD BY FINDING THE POINT ON THE LINE SEGMENT
C                 ACC(1,.) + ALPHA*(X - ACC(1,.)),    0<= ALPHA<= 1
C    THAT SATISFIES THE VIOLATED CONSTRAINT EXACTLY.
C
  170 JJ= KGMIN
      MPD= MP2
      MFLAG= 1
      WRITE (6,174) MPD,JJ,GMIN
  174 FORMAT(/,' THE QUADRATIC PICKED THE WRONG CONSTRAINT.',/
     1,'  THE MOST INFEASIBLE CONSTRAINT IS TYPE ',I3,
     2'  NUMBER ',I4,/,' WITH A VALUE OF',F14.6)
      DO 171 I= 1,DD
  171     V1(I)= X(I)- ACC(3,I)
      DO 175 I= 1,4
  175       DOT(I)= 0.0
      IF (MPD.EQ.0) GO TO 190
      IF (MPD.EQ.-1) GO TO 180
      DO 172 I= 1,KNU
         IK= I+ KMU
         DOT(1)= DOT(1)+ G1(JJ,I)*ACC(3,IK)
         DOT(2)= DOT(2)+ G1(JJ,I)*V1(IK)
  172 CONTINUE
      STR= (-BA(JJ)-DOT(1))/DOT(2)
      DO 173 I= 1,DD
  173    X(I)= ACC(3,I)+ STR*V1(I)
      GO TO 295
  180 DO 183 I= 1,KMU
         DOT(1)= DOT(1)+ G2(JJ,I)*ACC(3,I)
         DOT(2)= DOT(2)+ G2(JJ,I)*V1(I)
  183 CONTINUE
      STR= (-BB(JJ)-DOT(1))/DOT(2)
      DO 186 I= 1,DD
  186    X(I)= ACC(3,I)+ STR*V1(I)
      GO TO 295
C
C        THE BILINEAR CONSTRAINT IS NOT SATISFIED AT X.  FIND A GOOD
C     STARTING POINT BY SOLVING THE QUADRATIC DEFINED BY THE INTER-
C     SECTION OF THE SEGMENT DEFINED ABOVE AND THE CONSTRAINT.
C
  190 IF (INEQ.EQ.2) GO TO 200
      DO 192 I= 1,DD
         IF (I.GT.KMU) GO TO 191
         DOT(1)= DOT(1)+ D1(INQ,I)*ACC(3,I)
         DOT(2)= DOT(2)+ D1(INQ,I)*V1(I)
         GO TO 192
  191    IK= I- KMU
         DOT(3)= DOT(3)+ F1(INQ,IK)*ACC(3,I)
         DOT(4)= DOT(4)+ F1(INQ,IK)*V1(I)
  192 CONTINUE
      U(1)= -V1(INQ)*DOT(4)
      U(2)= -(V1(INQ)*E1(INQ)+ ACC(3,INQ)*DOT(4))+ DOT(2)
      U(3)= BE1(INQ)+ DOT(1)- ACC(3,INQ)*(DOT(3)+ E1(INQ))
      CALL QUADS(U,IMAG,STR,BETA)
      IF (STR.LT.TOLCV) STR= BETA
      DO 193 I= 1,DD
  193    X(I)= ACC(3,I)+ STR*V1(I)
      GO TO 295
  200 IB= INP+ KMU
      IF ( KMU.EQ.0) GO TO 204
      DO 202 I= 1,KMU
         DOT(1)= DOT(1)+ D2(INP,I)*ACC(3,I)
         DOT(2)= DOT(2)+ D2(INP,I)*V1(I)
         DOT(3)= DOT(3)+ F2(INP,I)*ACC(3,I)
         DOT(4)= DOT(4)+ F2(INP,I)*V1(I)
  202 CONTINUE
  204 U(1)= -V1(IB)*DOT(4)
      U(2)= -(V1(IB)*E2(INP)+ ACC(3,INP)*DOT(4))+ DOT(2)
      U(3)= BE2(INP)+ DOT(1)- ACC(3,INP)*(DOT(3)+ E2(INP))
      CALL QUADS(U,IMAG,STR,BETA)
      IF (STR.LT.TOLCV) STR= BETA
      DO 203 I= 1,DD
  203    X(I)= ACC(3,I)+ STR*V1(I)
      GO TO 295
  300 WRITE (6,980)
  850 FORMAT(1X,'THE CURRENT LINEAR APPROXIMATION TO THE CURVE IS:'
     1,/,1X,'              A2(*U)+                 A3')
  855 FORMAT (1X,2F17.7)
  936 FORMAT(1X,'NORM(F(X))= ',F10.6,'  F= ',9F10.6)
  940 FORMAT (1X,//'*****ALGORITHM BOMBED WITH SINGULAR JACOBIAN*****')
  945 FORMAT(1X,'NO DESCENT POSSIBLE -- ALPHA WAS ZERO.')
  980 FORMAT(1X,'NEWTON"S METHOD FAILED TO CONVERGE.')
      STOP
      END
C      QUADS FINDS THE SMALLEST NONNEGATIVE ROOT OF
C     U(1)*ALPHA**2 + U(2)*ALPHA + U(3) = 0. IF THERE
C     IS ONE.  IF NOT, IMAG IS SET = 1.
C
      SUBROUTINE QUADS(U,IMAG,ALPHA,BETA)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,SS,PR,ZFLAG,P,DD,DD1,BL1,BL2
      INTEGER*2 JH(350),DIGMA(552),KINBAS(1302),IDBAS(1302)
      COMMON/TOLER/ TOLEZ,TOLRD,TOLCV,STPMX,STPRD
      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
      DIMENSION U(3)
      IMAG= 0
      IF (DABS(U(1)).GT. TOLEZ) GO TO 10
      IF (DABS(U(2)).LT.TOLEZ) GO TO 8
      ALPHA = -U(3)/U(2)
      BETA= ALPHA
      RETURN
    8 IMAG= 1
      RETURN
   10 RT= U(2)**2 - 4*U(1)*U(3)
      IF (RT) 20, 30, 40
C
C      THE QUADRATIC DOES NOT INTERSECT THIS CONSTRAINT.
C
   20 IMAG= 1
      RETURN
C
C     THERE IS A DOUBLE ROOT.
C
   30 ALPHA = U(2)/(U(1)*2.)
      BETA= ALPHA
      RETURN
C
C      TWO REAL ROOTS EXIST.
C
   40 DIS= DSQRT(RT)
      ALPHA= (-U(2)-DIS)/(2.*U(1))
      BETA= (-U(2)+DIS)/(2.*U(1))
      IF (ALPHA .LE. BETA) RETURN
      SAVE= ALPHA
      ALPHA = BETA
      BETA= SAVE
      RETURN
      END
      SUBROUTINE CONCHK(GMIN,KGMIN,MP2)
C
C      THIS SUBROUTINE EVALUATES THE CONSTRAINT FUNCTIONS
C  AND FINDS THE SMALLEST VALUE IN GMIN.  ZFLAG IS BOTH AN
C  INPUT AND OUTPUT PARAMATER.  IF ZFLAG.EQ.1 INITIALLY, THEN
C  ONLY THE NONLINEAR CONSTRAINT IS EVALUATED.  ZFLAG IS SET
C  EQUAL TO 2 IF ANY CONSTRAINT IS NONPOSITIVE. OTHERWISE IT
```

```
C   REMAINS ZERO.
C
         IMPLICIT REAL*8 (A-H,O-Z)
         INTEGER PD,PD1,PD2,SS,RF,ZFLAG,P,DD,DD1,BL1,BL2
         INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(80))
         REAL   A(4000),C(800),CMIN,COND,ERMAX,SUMINF
         INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
         COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10)
         COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
         COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
         COMMON/LP1/PI(1302),XX(1302)
         COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
         COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
         COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
         COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
         GMIN= 10.
         ZFLAG= 0
         DO 20 I= 1,M
             IK= JH(I)
             GG= 0.0
             IF (KNU.EQ.0) GO TO 15
             DO 10 J= 1,KNU
                 JK= J+ KNU
                 GG= GG+ G1(I,J)*X(JK)
   10        CONTINUE
   15        GG= GG+ BA(I)
             XX(IK)= GG
             IF (IK .EQ. N) GO TO 20
             IF (GG.GE.GMIN) GO TO 20
             GMIN= GG
             KGMIN= I
             MP2= 1
   20    CONTINUE
         DO 40 I= 1,N
             IK= DIGMA(I)
             GG= 0.0
             IF (KMU.EQ.0) GO TO 35
             DO 30 J= 1,KMU
                 GG= GG+ G2(I,J)*X(J)
   30       CONTINUE
   35        GG= GG+ BB(I)
             PI(IK)= GG
             IF (GG.GE.GMIN) GO TO 40
             GMIN= GG
             KGMIN= I
             MP2= -1
   40    CONTINUE
         IF (PD.EQ.0) RETURN
   45    DOT= 0.
         DOT1= 0.
         GG= 0.0
         IF (INEQ - 2) 50, 60, 60
   50    IF (KNU.EQ.0) GO TO 55
         DO 54 J= 1,KNU
         JK= J+ KMU
   54    DOT= DOT+ F1(INQ,J)*X(JK)
   55    GG= BF1(INQ)- X(INQ)*(DOT+ E1(INQ))
         IF (KMU.EQ.0) GO TO 80
         DO 57 J= 1,KMU
   57    GG= GG+ D1(INQ,J)*X(J)
```

```
      GO TO 80
   60 KB2= KMU+ INP
      IF (KMU.EQ.0) GO TO 65
      DO 63 J= 1,KMU
      DOT= DOT+ F2(INP,J)*X(J)
   63 DOT1= DOT1+ D2(INP,J)*X(J)
   65 GG= BF2(INP)+ DOT1- X(KB2)*(DOT+ E2(INP))
      GO TO 80
   80 IF (GG.GE.GMIN) RETURN
      GMIN= GG
      MP2= 0
      RETURN
      END
C
C       SUBROUTINE FTN EVALUATES THE FUNCTION F(X)=
C         <B(MU)+ D1*X(MU)  + DIAG(X(MU))*(F1*X(NU))          >
C         <B(NU)+ D2*X(MU)  + DIAG(X(NU))*(F2*X(MU)+ F3)      >
C
      SUBROUTINE FTN(F,Y)
C
C         THIS ROUTINE EVALUATES THE FUNCTION WHICH THE ENDPOINT
C     SUBROUTINE IS CURRENTLY TRYING TO FIND A ROOT OF.  IF MFLAG
C     EQUALS 1 THEN THE LAST FUNCTIONAL COMPRISING F IS ONE OF THE
C     INEQUALITY CONSTRAINTS-- THE INEQUALITY DETERMINED BY THE
C     PARAMETERS MPD AND JJ.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,P,DD,DD1,BL1,BL2
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IOBAS(1302)
      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
      INTEGER BLI,BLJ,B2L,CDC
      DIMENSION F(DD),Y(DD)
      F(DD)= 0.
      KFUN= KFUN+ DD
      KMU1= KMU+ 1
      IRO= 0
      IF (KMU.EQ.0) GO TO 23
      DO 20 I= 1,KMU
      IF (INEQ.EQ.2) GO TO 5
      IF (I.EQ.INQ) GO TO 19
    5 IR= I- IRO
      DOT= 0.0
      IF (KNU.EQ.0) GO TO 15
      DO 10 J= 1,KNU
      JK= J+ KMU
   10 DOT= DOT+ F1(I,J)*Y(JK)
   15 F(IR)= -Y(I)*(DOT+ E1(I))+ BF1(I)
      IF(KMU.EQ.0) GO TO 20
      DO 18 J= 1,KMU
   18 F(IR)= F(IR)+ D1(I,J)*Y(J)
      GO TO 20
   19 IRO= 1
   20 CONTINUE
   23 IF (IBL2.EQ.0) GO TO 70
      IB= INP+ KMU
      DO 40 I= KMU1,DD
      IMB= I- KMU
      IF (INEQ.EQ.1) GO TO 24
      IF (I.EQ.IB) GO TO 35
   24 IR= I- IRO
      DOT= 0.0
      IF(KMU.EQ.0) GO TO 30
      DO 25 J= 1,KMU
   25 DOT= DOT+ F2(IMB,J)*Y(J)
   30 F(IR)= -Y(I)*(DOT+E2(IMB))+ BF2(IMB)
      IF (KMU.EQ.0) GO TO 40
      DO 32 J= 1,KMU
   32 F(IR)= F(IR)+ D2(IMB,J)*Y(J)
      GO TO 40
   35 IRO= 1
   40 CONTINUE
   70 IF (MFLAG.LT.1) RETURN
      IF (MPD.NE.0) GO TO 120
      DOT= 0.
      DOT1= 0.
      DOT2= 0.
      IF (INEQ - 2) 80, 90,  90
   80 IF (KNU.EQ.0) GO TO 85
      DO 84 J= 1,KNU
      JK= J+ KMU
   84 DOT= DOT+ F1(INQ,J)*Y(JK)
   85 F(DD)= BF1(INQ)- Y(INQ)*(DOT+ E1(INQ))
      IF (KMU.EQ.0) GO TO 120
      DO 87 J= 1,KMU
   87 F(DD)= F(DD)+ D1(INQ,J)*Y(J)
      GO TO 120
   90 IF (KMU.EQ.0) GO TO 95
      DO 93 J= 1,KMU
      DOT= DOT+ F2(INP,J)*Y(J)
   93 DOT1= DOT1+ D2(INP,J)*Y(J)
   95 IKN= INP+ KMU
      F(DD)= BF2(INP)+ DOT1- Y(IKN)*(DOT+ E2(INP))
  120 IF (MFLAG.LT.1) RETURN
      IF (MPD.EQ.0) RETURN
      IF (MFLAG - 2) 125,160,170
  125 IF (MPD.EQ. -1) GO TO 140
      IF (KNU.EQ.0) GO TO 135
      DO 130 J= 1,KNU
      JK= J+ KMU
  130 F(DD)= F(DD)+ G1(JJ,J)*Y(JK)
  135 F(DD)= F(DD)+ BA(JJ)
      RETURN
  140 IF (KMU.EQ.0) GO TO 150
      DO 145 J= 1,KMU
  145 F(DD)= F(DD)+ G2(JJ,J)*Y(J)
  150 F(DD)= F(DD)+ EB(JJ)
      RETURN
  160 F(DD)= Y(JJ)
      RETURN
  170 DOT= 0.0
      DO 180 J= 1,DD
  180       DOT= DOT+ ACC(2,J)*Y(J)
```

```
      F(DD)= DOT- BT
      RETURN
      END
      SUBROUTINE DERIV
C
C     THIS SUBROUTINE CALCULATES THE EXACT JACOBIAN OF THE
C     BILINEAR FUNCTION DEFINED BY BLCONS.  THERE MAY OR MAY
C     NOT BE ANOTHER FUNCTIONAL APPENDED WHICH WE ARE ATTEMPT-
C     ING TO MAKE BINDING WITH NEWTON'S METHOD.  ALSO, A COL-
C     UMN MAY BE OMITTED IF THE ENDPOINT ALGORITHM REQUIRES
C     THE CORRESPONDING VARIABLE TO BE FIXED.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,SS,RR,ZFLAG,P,DD,DD1,BL1,BL2
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10)
      COMMON/BLCST/BB1(10),BB2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
      INTEGER COLJJ
      KJAC= KJAC+ DD * DD
      IF (MFLAG.EQ.0) KJAC= KJAC- DD
      DO 10 I= 1,DD
         DO 10 J= 1,DD
   10       H(I,J)= 0.
      KMU1= KMU+1
      KNU1= KNU+ 1
      DD1= DD-1
      COLJJ= 0
      IF (KMU.EQ.0) GO TO 55
      DO 50 J=1,KMU
         IF (J.EQ.JJ .AND. MFLAG.GT.1) GO TO 45
         JM= J-COLJJ
      IF (KMU.EQ.0) GO TO 35
         IRC= 0
         DO 30 I= 1,KMU
            IF (INEQ.EQ.2) GO TO 15
            IF (I.EQ.INQ) GO TO 28
   15       IR= I- IRC
            H(IR,JM)= D1(I,J)
            IF (I.NE.J) GO TO 30
            DOT= 0.0
            IF (KNU.EQ.0) GO TO 25
            DO 20 K= 1,KNU
            KB= KMU+K
   20          DOT= DOT+ F1(I,K)*X(KB)
   25       H(IR,JM)= H(IR,JM)- DOT- E1(I)
            GO TO 30
   28       IRC= 1
   30    CONTINUE
   35    IF (BL2.EQ.0) GO TO 50
         DO 40 I= KMU1,DD
         IMB= I- KMU
            IF(INEQ.EQ.1) GO TO 37
            IF(IMB.EQ.INP) GO TO 38
   37               IR= I- IRC
                    H(IR,JM)= D2(IMB,J)- F2(IMB,J)*X(I)
                    GO TO 40
   38               IRC= 1
   40          CONTINUE
               GO TO 50
   45          COLJJ= 1
   50     CONTINUE
   55     IF (KNU.EQ.0) GO TO 141
          DO 70 J= KMU1,DD
               IF(J.EQ.JJ .AND. MFLAG.GT.1) GO TO 65
               JM= J-COLJJ
               JMU= J-KMU
          IF (KMU.EQ.0) GO TO 72
               IRC= 0
               DO 60 I= 1,KNU
                    IF (INEQ.EQ.2) GO TO 57
                    IF (I.EQ.INQ) GO TO 58
   57               IR= I- IRC
                    H(IR,JM)= - X(I)*F1(I,JMU)
                    GO TO 60
   58               IRC= 1
   60          CONTINUE
               GO TO 70
   65          COLJJ= 1
   70     CONTINUE
          COLJJ= 0
   72     IRC= 0
          IF (INEQ .EQ. 1) IRC= 1
          IF (BL2.EQ.0) GO TO 141
          DO 80 I= 1,KNU
          IKM= I+KMU
          IF (IKM.GE.JJ.AND.MFLAG.GT.1) COLJJ= 1
          IF (IKM.EQ.JJ .AND. MFLAG.GT.1) GO TO 80
          IF (INEQ.EQ.1) GO TO 73
          IF (I.EQ.INP) GO TO 78
   73     IR= I- IRC
          IK= KMU+IR
          KNC= IKM - COLJJ
          IF (KMU.EQ.0) GO TO 77
          DO 75 K= 1,KMU
   75     H(IK,KNC)= H(IK,KNC)- F2(I,K)*X(K)
   77     H(IK,KNC)= H(IK,KNC)- F2(I)
          GO TO 80
   78     IRC= 1
   80     CONTINUE
C
C            IF MFLAG=1 THE INEQUALITY SPECIFIED BY MPD AND JJ
C        DETERMINES THE DDTH ROW OF THE JACOBIAN MATRIX H.
C
  141     IF (MFLAG.LT.1) RETURN
          IF (MPD.EQ.0) GO TO 145
          GO TO 190
  145     IF (INEQ-2) 150,160,160
  150     DO 155 J= 1,DD
          IF (J.GT.KMU) GO TO 157
          H(DD,J)= D1(INQ,J)
          IF (J.NE. INQ) GO TO 155
          DOT = 0.0
          IF (KNU.EQ.0) GO TO 154
```

```
            DO 152 K= 1,KNU
            KB= KMU+ K
        152 DOT= DOT+ F1(INQ,K)*X(KB)
        154 H(DD,J)= H(DD,J)- DOT- E1(KMU)
            GO TO 155
        157 JK= J-KMU
            H(DD,J)= -F1(INQ,JK)*X(INQ)
        155 CONTINUE
            GO TO 180
        160 KB2= INP+ KMU
            DO 164 J= 1,DD
            IF (J.GT. KMU) GO TO 165
            H(DD,J)= D2(INP,J)- F2(INP,J)*X(KB2)
            GO TO 164
        165 IF (J.NE.KB2) GO TO 164
            DOT= 0.0
            IF (KMU.EQ.0) GO TO 168
             DO 167 K= 1,KMU
        167 DOT = DOT+ F2(INP,K)*X(K)
        168 H(DD,J)= -DOT-E2(INP)
        164 CONTINUE
            GO TO 180
        180 IF (MFLAG.LT.1)  RETURN
            IF (MPD.EQ.0) RETURN
            IF (MFLAG - 2) 190,230,240
        190 IF (MPD.EQ. -1) GO TO 210
            DO 200 J= KMU1,DD
            JM= J- KMU
        200 H(DD,J)= G1(JJ,JM)
            RETURN
        210 DO 220 J= 1,KMU
        220 H(DD,J)= C2(JJ,J)
            RETURN
        230 H(DD,JJ)= 1.0
            RETURN
        240 DO 250 I= 1,DD
        250     H(DD,I)= ACC(2,I)
            RETURN
            END
            SUBROUTINE NORM(Y,S1,N)
            IMPLICIT REAL*8(A-H,O-Z)
            DIMENSION Y(10)
            S1= 0.0
            DO 10 I= 1,N
         10     S1= S1+ Y(I)*Y(I)
            S1= DSQRT(S1)
            RETURN
            END
      C
      C      DSENT IS A SUBROUTINE WHICH IMPLEMENTS DESCENT ON
      C      THE NORM OF THE FUNCTION WHOSE ROOT IS BEING DETER-
      C      MINED BY NEWTON'S METHOD.
      C
            SUBROUTINE DSENT(ALPHA,FNORM)
            IMPLICIT REAL*8 (A-H,O-Z)
            INTEGER PD,PD1,PD2,SS,RF,ZFLAG,P,DD,DD1,BL1,BL2
            COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
            COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10)
            DIMENSION Y(10),F(10)
          1 DO 5 I=1,DD
```

```
    5 Y(I)= X(I)- ALPHA*Z(I)
      CALL FUN(F,Y)
      CALL NORM(F,S2,DE)
      IF (S2.LT.PNORM) RETURN
      ALPHA= ALPHA/2.
      IF (ALPHA.GT. .00001) GO TO 1
      RETURN
      END
C
C       DECOMP AND SOLVE ARE BORROWED FROM FORSYTHE'S THE SOLUTION
C    OF LINAER SYSTEMS OF EQUATIONS.  THEY IMPLEMENT A GAUSSIAN ELIM-
C    INATION SCHEME WITH PARTIAL PIVOTING TO PRODUCE A LU DECOMPOSTION
C    OF THE MATRIX A.  'SOLVE' PERFORMS THE BACK SUBSTITIONS NECESSARY
C    TO SOLVE     UL * X = B .  KDET FINDS SIGN( DET A ).
C
      SUBROUTINE DECOMP(NN,A,UL)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/INT/ IPS(30),KDET,KOUNT,ISING
      DIMENSION A(10,10),UL(10,10),SCALES(10)
      N= NN
      KDET= 1
C
C    INITIALIZE IPS,UL, AND SCALES.
      ISING= 0
      DO 5 I= 1,N
         IPS(I)= I
         ROWNRM= 0.0
         DO 2  J= 1,N
             UL(I,J)= A(I,J)
             IF (ROWNRM-DABS(UL(I,J))) 1,2,2
    1        ROWNRM= DABS(UL(I,J))
    2    CONTINUE
         IF (ROWNRM) 3,4,3
    3    SCALES(I)= 1./ROWNRM
         GO TO 5
    4    CALL SING(1)
         ISING= 1
         SCALES(I)= 0.
    5 CONTINUE
C
C    GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
      NM1= N- 1
      DO 17 K= 1,NM1
         BIG= 0.0
         DO 11  I= K,N
             IP= IPS(I)
             SIZE= DABS(UL(IP,K))*SCALES(IP)
             IF (SIZE-BIG)  11,11,10
   10                BIG= SIZE
                     IDXPIV= I
   11    CONTINUE
         IF (BIG)  13,12,13
   12        CALL SING(2)
             ISING= 1
             GO TO 17
   13    IF (IDXPIV-K) 14,15,14
   14        J= IPS(K)
             IPS(K)= IPS(IDXPIV)
             IPS(IDXPIV)= J
             KDET= -KDET
```

```
 15       KP= IPS(K)
          PIVOT= UL(KP,K)
          KP1= K+1
          DO 16 I= KP1,N
              IP= IPS(I)
              EM= -UL(IP,K)/PIVOT
              UL(IP,K)= -EM
              DO 16 J= KP1,N
                  UL(IP,J)= UL(IP,J)+ EM*UL(KP,J)
 16       CONTINUE
 17 CONTINUE
    KP= IPS(N)
    IF (UL(KP,N)) 19,18,19
 18 CALL SING(2)
    ISING= 1
 19 RETURN
    END
C
C
    SUBROUTINE SOLVE(NN,UL,B,X)
    IMPLICIT REAL*8(A-H,O-Z)
    COMMON/INT/ IPS(30),KDET,KCUNT,ISING
    DIMENSION UL(10,10),B(10),X(10)
    N= NN
    NP1= N+ 1
    IP= IPS(1)
    X(1)= B(IP)
    DO 2  I= 2,N
          IP= IPS(I)
          IM1= I-1
          SUM= 0.0
          DO 1 J= 1,IM1
  1           SUM= SUM+ UL(IP,J)*X(J)
  2 X(I)= B(IP)- SUM
C
    IP= IPS(N)
    X(N)= X(N)/UL(IP,N)
    IF(UL(IP,N).GE.0) GO TO 10
          KDET= -KDET
 10 DO 4 IBACK= 2,N
    I= NP1- IBACK
C   I GOES (N-1).....1
          IP= IPS(I)
          IP1= I+1
          SUM= 0.0
          DO 3 J= IP1,N
  3           SUM= SUM+ UL(IP,J)*X(J)
          DIV= UL(IP,I)
          IF(DIV.GE.0) GO TO 4
          KDET= -KDET
  4 X(I)= (X(I)-SUM)/DIV
    RETURN
    END
C
C
    SUBROUTINE SING(IWHY)
 11 FORMAT(1X,'MATRIX WITH ZERO ROW IN DECOMPOSE.')
 12 FORMAT(1X,'SINGULAR MATRIX IN DECOMPOSE. ZERO DIVIDE IN SOLVE')    21
    IF (IWHY- 1) 1,1,2
  1 WRITE (6,11)
```

```
      GO TO 10
    2 WRITE (6,12)
   10 RETURN
      END
      SUBROUTINE DEBUG(MODE)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MIN
      REAL*4 TIME
      INTEGER PD,PD1,PD2,SS,RP,ZFLAG,RS,P,DD,DD1,BL1,BL2
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
     1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFFZ,JCCLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NUECI,NDUAL,NIPIW,IFEAS,IFCRSH
      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
      COMMON IA(4000),IE(8000)
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/SCAL/ BT,NB,JJ,MFLAG,DD1,P,PD,MPD,INEQ,KFUN,KJAC
      COMMON/DIM/ IH,N,M,DD,KMU,KNU,BL1,BL2,MP1,NM,INQ,INP
      IF (MODE-2) 10,30,60
   10 CALL LEFT1A(TIME)
      WRITE(6,20) TIME
   20 FORMAT (1X,' THE TIME LEFT IS NOW ',F9.6,' SEC.')
      RETURN
   30 WRITE (6,40)
      WRITE (6,41)  (BA(I),I= 1,M)
      WRITE (6,49) (BB(I),I=1,N)
      IF (KNU.EQ.0) GO TO 35
      WRITE (6,42) ((G1(I,J),I= 1,M),J= 1,KNU)
   35 IF (KMU.EQ.0) RETURN
      WRITE (6,43) ((G2(I,J),I= 1,N),J= 1,KMU)
      RETURN
   40 FORMAT (/,1X,' LINEAR CONSTRAINTS.')
   42 FORMAT (1X,' G1= ',7F13.5)
   41 FORMAT (1X,' BA= ',7F13.5)
   49 FORMAT (1X,' BB= ',7F13.5)
   43 FORMAT (1X,' G2= ',7F13.5)
   60 IF (MODE - 4) 65,70,100
   65 WRITE (6,80)  (KINBAS(I),I=1,NCOL)
      WRITE (6,85) (JH(I),I= 1,NROW)
   80 FORMAT (1X,' KINBAS= ',20I4)
   85 FORMAT (1X,' JH= ',20I4)
      RETURN
   70 WRITE (6,71) KNU,KMU
      WRITE (6,152) (NUH(I),I=1,IH)
      WRITE (6,151) (MUH(I),I=1,IH)
      WRITE (6,74) (XX(I),I=1,IH)
      WRITE (6,75) (PI(I),I=1,IH)
```

```
 71 FORMAT (//,' SUPERBASIC INFO: KNU=',I3,'  KMU= ',I3)
 74 FORMAT ('   T    =',9F12.6)
 75 FORMAT (' LAMBDA=',9F12.6)
    RETURN
100 IF (MODE - 6)  101,140,140
101 WRITE (6,120)  KNU,KMU
    IF (KMU.EQ.0) GO TO 105
    WRITE (6,121) (BF1(I),I=1,KMU)
    WRITE (6,122) ((D1(I,J),J=1,KMU),I=1,KMU)
    WRITE (6,123) (E1(I),I=1,KMU)
    IF (KNU.EQ.0) GO TO 105
    WRITE (6,124) ((F1(I,J),J=1,KNU),I=1,KMU)
105 IF (KNU.EQ.0) RETURN
    WRITE (6,125) (BF2(I),I=1,KNU)
    WRITE (6,126) (E2(I),I=1,KNU)
    IF (KMU.EQ.0) RETURN
    WRITE (6,127) ((C2(I,J),J=1,KMU),I=1,KNU)
    WRITE (6,128) ((F2(I,J),J=1,KMU),I=1,KNU)
    RETURN
120 FORMAT (1X,'BL CONSTRAINTS.  KNU,KMU:',2I3)
121 FORMAT (1X,' BF1= ', F14.6)
122 FORMAT (1X,' D1= ',F14.6)
123 FORMAT (1X,' E1= ',F14.6)
124 FORMAT (1X,' F1= ',F14.6)
125 FORMAT (1X,' BF2=',F14.6)
126 FORMAT (1X,' E2= ',F14.6)
127 FORMAT (1X,' C2= ',F14.6)
128 FORMAT (1X,' F2= ',F14.6)
140 WRITE (6,150) PD,KMU,KNU,INEQ
    WRITE (6,151) (MUH(I),I= 1,IH)
    WRITE (6,152) (NUH(I),I= 1,IH)
150 FORMAT (//,1X,' ENDPOINT FINDER DATA:PD,KMU,KNU,INEQ= ',4I3)
151 FORMAT (1X,'MUH(I)= ',20I3)
152 FORMAT (1X,'NUH(I)= ',20I3)
    WRITE (6,154) (JH(I),I=1,M)
    WRITE (6,155) (DIGMA(I),I=1,N)
154 FORMAT (1X,'JH= ',15I4)
155 FORMAT (1X,'DIGMA= ',15I4)
    WRITE (6,153) (XX(I),I=1,NM)
    WRITE (6,156) (PI(I),I=1,NM)
153 FORMAT (1X,' XX= ',9F13.6)
156 FORMAT (1X,' PI= ',9F13.6)
    RETURN
    END
```

III. The HRA (Homotopy Retraction Algorithm) Code for Solving Equilibrium Problems

III.1. Revisions to the Original Code

The homotopy retraction algorithm is described in detail in Chapter IV of the report [2]. However, since the completion of that work the author has changed the HRA code to improve the domain of convergence of the code with respect to the user supplied initial utility levels. With these changes the code successfully finds equilibrium points for problems and starting points which resulted in failure for the previous version of the HRA. Also, solution times have been as fast, or faster, than before. This version of the code has not been tested on a sufficient number of test problems but the advantages over the old code seem to be so great that we should describe the code in its present form.

By Lemma II.2.4, solving the equilibrium problem is equivalent to solving for

$$w \equiv (\pi,\ \lambda,\ \zeta^i(i \in \underline{m}),\ p;\ s,\ t,\ z^i(i \in \underline{m}),\ p)$$

which satisfies

$$f(w) = 0, \qquad w \in D\ , \tag{5}$$

where

$$f(w) = \begin{pmatrix} \pi(w)w^1 - \lambda_1(w)(t_1(w) + v_1) \\ \vdots \\ \pi(w)w^m - \lambda_m(w)(t_m(w) + v_m) \end{pmatrix}$$

($\pi(w)$ refers to the first m components of w, etc., and

$$\begin{array}{ll}
D \equiv \{w \mid \gamma^i Z^i - t_i = v_i & i \in \underline{m} \\[2ex]
\sum_{i=1}^{m} B^i Z^i + pe + s = \sum_{i=1}^{m} w^i & \\[2ex]
\pi B^i - \lambda_i \gamma^i - \zeta^i = 0 \, , & i \in \underline{m} \\[2ex]
\pi e - p = 1 & \\[2ex]
\pi s = 0 & \\[2ex]
\zeta^i z^i = 0 \, , & i \in \underline{m} \\[2ex]
w \geq 0 & \} \, .
\end{array}$$

The original homotopy which is used to define a path to the solution of (1.2) is

$$F(w, \theta) = \theta f(w) - (1 - \theta)(\lambda(w) - \lambda^0)$$

where λ^0 is determined by the solution to the auxilliary linear program (2). As part of the modification to HRA we changed the deformation above to

$$F(w, \theta) = \theta f(w) - (1 - \theta)(t(w) - t^0) \quad .$$

Here $t(w)$ refers to the slack variables of the first m rows, where m is the number of consumers in the economy. By the properties of the auxiliary linear program, it will always be true that t^0, the value of $t(w)$ at the optimum, is identically zero. This is because one can never achieve a higher level of exports p by allowing consumers to have a greater utility level than v_i. Thus, we can simplify the definition of the homotopy to $F:D\chi[0,1] \to R^m$,

$$F(w, \theta) = \theta f(w) - (1 - \theta)t(w) \quad . \tag{6}$$

A convergence theorem using the path defined by (6) can be proved even though we will not do so here.

Other amendments to the algorithm include a procedure for altering the initial utility levels v_i, $i = 1, \ldots, m$, if the processing of the auxiliary linear program indicates that the solution is unsatisfactory. There are three factors which can cause the program to change the initial utility levels:

1. If the l.p. is infeasible, then v_i is too large. They will all be reduced.
2. If some $\lambda_i^0 = 0$ initially, it indicates that v_i can be increased without changing the optimal value of the objective function. For each i for which this is true we increase v_i until $\lambda_i > 0$.

3. If $f_i(w) \leq 0$ for some i, the initial boundary conditions are not satisfied. By reducing v_i for such indices i, we can increase f_i until it is positive.

When we say to increase v_i or decrease v_i, we mean that the following procedure is followed:

a) After each solution of the auxiliary LP, a scalar TI is incremented by one starting with TI = 5.

b) If any of the three conditions above occurs, to reduce v_i we let

$$v_i = v_i*(1 - 1/TI) ,$$

to increase v_i we let

$$v_i = v_i*(1 + 1/TI) .$$

c) When TI reaches 20 and the solution to the LP is still unsatisfactory, the program terminates.

III.2. Input Requirements

The form of the input data for the HRA is exactly identical to the form for the BCA. See Section II.1 for a description.

III.3. Main Program

The main program consists of 390 source statements which perform many of the same functions as the main program for the BCA. The parameters are initialized and possibly altered by reading the namelist PARM1. The subroutines of LPM1 are called to input and solve the auxiliary linear program. Tests 1 and 2 of Section 1 are performed. The data is read so the C matrix can be constructed. Then the budget surpluses f_i are calculated and test 3 is performed. The rest of the main program is essentially identical to that of the BCA except that different decisions are made after the ENDPNT subroutine has been called to determine which is the next cell (see Figure IV.3.1, Elken [1977b]).

Restrictions Relevant to the Use of HRA

1. The number of consumers (IH) must be less than or equal to 10.
2. The matrix D must not have more than 350 rows or 400 columns or more than 4000 non-zero elements.

Of course, if one must solve a problem larger than these dimensions allow, more core must be allocated and the dimensions of the appropriate variables must be changed in every subroutine in which they appear.

Before describing the other subroutines of HRA we will define the variables which differ from those in BCA as described in Section II.2.

Variable Glossary

In HRA we refer to one more variable from the blank common of LPMI.

$$\text{MSTAT} = \begin{cases} \text{'N'} & \text{if the auxiliary linear program} \\ & \text{is infeasible} \\ \text{'O'} & \text{if the current solution is optimal} \\ \text{'U'} & \text{if the LP is unbounded.} \end{cases}$$

We use MSTAT to check if the current auxiliary linear program is infeasible.

The common blocks LP1, BLCST, BLCST2, LNCONS, INDX1, and INDX2 are identical in both HRA and BCA. COMMON/INT/ contains the variable KEND, the number of times the ENDPNT subroutine is called.

We include this variable to pass its value from the main program to ENDPNT so that we can test to see whether this is the first call to ENDPNT or not. In addition to those variables described in Section II.2, COMMON/SCAL/ contains the variable

IHP1 = IH + 1 the number of consumers plus one. This is the dimension of the consumer space plus one for the homotopy parameter.

COMMON/DIM/

$$\text{ITAIL} = \begin{cases} \text{1, if the Newton tail routine is in effect,} \\ \text{0, otherwise} \end{cases}$$

The description of the algorithm in Chapter IV of Elken [2], shows when ITAIL is set equal to 1.

The rest of the common blocks are identical to those in BCA.

III.4. Subroutines of HRA

We will describe only the differences between the subroutines of HRA and those of BCA.

1) Subroutine BLCONS: Calculates the coefficients of the bilinear functionals the budget surpluses so that f(w), (6), can be evaluated in terms of the superbasic variables (see VI (4, 9) of Elken [2]. This subroutine is substantially the same as BLCONS in BCA except that in this case all the bilinear functionals are computed rather than the first DD, the number of bilinear constraints which are currently satisfied.

Subroutines called: UPAKC

2) SUBROUTINE FINDP (PD1, IS, P): same as in BCA. See Section II.3.

3) SUBROUTINE PIVOT (SS, RR), SUBROUTINE UPAKC (II), SUBROUTINE BSCNG (S, R), SUBROUTINE SUPERB (KEY, PK1, IS, (PD2, JS), and SUBROUTINE RECALC: same as in BCA, see Section II.3.

4) SUBROUTINE ENDPNT (JS, PD1, IS, NET): This routine implements the path-following algorithm described in Section IV.2 of [2]. The reader should consult that work to be able to understand this subroutine. Here we will describe some details which were not mentioned in Section II.3.

The independent variables (superbasics) are placed in a vector X of dimension IH + 1 = IHP1. The extra dimension is for the homotopy parameter, θ. Theta always occupies the last position in X, X(IHP1).

From computational experience, we know that the curve $F^{-1}(0)$ is highly nonlinear in the last variable when the algorithm begins. That is, for the first tangential approximation in the first cell, θ increases very quickly, initially, but quite slowly thereafter. Remember, θ increases from 0 to 1 as the algorithm progresses, although some decreases are possible. To remedy the poor guess at θ which would result from following the initial tangent all the way to the opposite boundary of the cell, on the first iteration we move only one quarter of the distance to the opposite boundary. This rather ad hoc procedure seems to work quite well. After this initial step, the algorithm described in the work cited is followed precisely. The first tangent of the first cell is characterized by the fact that KEND = 1 (KEND is the number of cells traversed) and INFL = 0 (INFL is set equal to 1 after the first steplength is reduced).

In this program, the integer variable MFLAG has a different meaning than in the BCA code

$$\text{MFLAG} = \begin{cases} 0 & \text{when the tangent to the curve is being computed,} \\ 1 & \text{when a "hyperplane subproblem" is being solved,} \\ 2 & \text{when an endpoint in the opposite facet is being} \\ & \text{solved for, and} \\ 3 & \text{when the equilibrium point appears to be in the} \\ & \text{current cell. Newton's method will be implemented} \\ & \text{as a tail routine on } F(X) = 0. \end{cases}$$

SUBROUTINES CALLED:

QUADS, DSENT, CONCHK, GTN, FTN, DERIVG, DERIV, NORM, DECOMP, SOLVE.

Subroutines QUADS (U, IMAG, ALPHA, BETA) and DSENT (ALPHA, PNORM) are the same as in BCA.

SUBROUTINE GTN (IFL, Q, U, F) evaluates the homotopy $F(w, \theta)$. When one writes this function in terms of the superbasic variables one has

$$\text{THETA} = \text{X(IHP1)}$$

$$\text{Q(X)} = \text{THETA*f(X)}$$

$$-(1 - \text{THETA})*\begin{pmatrix} \sum_{I=1}^{KNU} \text{G1(MUH(J), I)*X(KMU + I) + BA(MUH(J))}, & J = 1,\ldots,\text{KMU} \\ \\ \text{X(KMU + J)}\ , & J = 1,\ldots,\text{KNU} \end{pmatrix}$$

The values of the variables of t indexed by MU (the first KMU in the vector above) are saved in the vector BLAM(10). These values are used in the subroutine DERIVG to be described later.

The last component of Q contains the value of a functional determined by the value of IFL (MFLAG in subroutine ENDPNT). If IFL = 1, then

$$\text{Q(IHP1)} = \sum_{I=1}^{IH} \text{ACC(2, I)*(X(I) - ACC(1, I))} \quad .$$

If IFL = 2, then

$$Q(IHP1) = \begin{cases} \sum_{I=1}^{KNU} G1(JJ,\ I)*X(KMU + I) + BA(JJ)\ , & \text{if}\quad MPD = 1\ , \\ \\ \sum_{I=1}^{KMU} G2(JJ,\ I)*X(I) + BB(JJ)\ , & \text{if}\quad MPD = -1\ . \end{cases}$$

The subroutine FTN is called to evaluate f(X).

SUBROUTINE DERIVG (IFL, U, G, F): calculates the jacobian of the function evaluated by GTN. The jacobian is of the form

$$DG(X) = \left[[THETA*Df(X) - (1. - THETA) \left(\begin{array}{c|c} 0 & G1_{MUH,\cdot} \\ \hline 0 & I \end{array} \right) \middle| \ f(X) + \left(\begin{array}{c} BLAM \\ \hline X_{NU} \end{array} \right) \right]$$

SUBROUTINE CONCHK (GMIN, GKMIN, MP2): this subroutine evaluates the basic variables in both the primal and dual systems, saves the value of the minimum in GMIN and points to which variable it is with (KGMIN, MP2). From the theory of the homotopy retraction algorithm, it is known that the first IH primal and dual variables cannot become negative so we ignore these variables when searching for the minimum. Also, the primal and dual variables corresponding to the objective function are of no importance to the algorithm, so we do not compare these values when searching for the minimum.

SUBROUTINE FTN(F, Y): evaluates the bilinear budget surpluses and stores them in F. This subroutine is identical to that in the BCA code except that no additional functionals are evaluated. That task is performed by GTN as described above.

SUBROUTINE DERIV: evaluates the jacobian of the function described in FTN. This is identical to the subroutine by the same name in the BCA code except, again, it is simpler because no rows need be appended because of an additional functional.

SUBROUTINES NORM (Y, S1, NY, DECOMP (NN, A, UL), SOLVE (NN, UL, B, X), SING (IWHY), DEBUG (MODE): are identical to the subroutines with the same names in BCA.

III.5. Sample Problems

Below we give the output of the HRA program for the same two problems (MAS-Colell and Whisman) presented in Section II.4. The form of the input is identical for HRA and BCA so we will not repeat the input decks here.

The output is very similar. The reader will note that the subproblem solved by ENDPNT is (IH + 1)-dimensional with HRA while the dimension grew from 1 to IH for the BCA. On these two problems, though, the HRA code required fewer calls to ENDPNT before equilibrium is reached. This behavior is typical with larger problems as well.

```
              IEF375I  JCB /TREW8666/ START 77201.1245        STOP  77201.1245 CPU   0MIN 00.17SEC
 &PARM1
 TOLFZ=  .10000000000000000000D-09,TOLBD=  .10000000000000000001D-03,TOLCV=  .99999999999999999997D-05,TOLEQ=  .9999
55D-06,
 ICNTRL=            1,IECHO=             0,STPMX=  10.000000000000000000     ,STPRD=  .50000000000000000000     ,ICECHO=
H=
          3,L=            2,IFROD=             0,KOUTB=             0,ITLIME=          100
 &END
   IECHO,IH,ICECHC    0    3    1
 &PARAM
 ZTOLZE=  .100000005E-03,ZTOLPV=  .100000034E-05,ZTCOST=  .100000005E-03,ZSCALE= -1.00000000      ,NRMAX=
      2000,
 NEMAX=        8000,IALG=              0,NDEGI=              1,NDUAL=          999,NIPIW=          99999,KINP=            5,IFB
 0,IFCRSH=
          1,IOBJ=            6,ITCH=             30,INVFRQ=          30,ITRLIM=         99999,ZTOLRP=  .999999975E-05,Z
000016E-01,
 ZTETA=  .100000043E-11,KINB=             0,ZTOLDA=  .100000008E-08
 &END
 PROBLEM  MAS-COL                                                                                    99999999999

 PROBLEM STATISTICS                                                                                            1,I
     6 ROWS
     4 STRUCTURAL COLUMNS
    12 NON-ZERO ELEMENTS
 DENSITY = 0.50000
  THE TIME LEFT IS NOW 14.948434 SEC.                                                                  350,NTMAX=
 INVERT STATISTICS
   14 NONZ IN BASIS                                                                                   AS=
    4 STRUCTURAL COLUMNS IN BASIS
    3 VECTORS ABOVE BUMP                                                                              TOLRI=  .100
    3 VECTORS BELOW BUMP
 L:     9 NONZ     3 ETAS
 U:     3 NONZ     1 ETAS
 TOTALS:     8 OFF DIAG NONZ     4 ETAS
 RELATIVE ERROR IN X =     0.63725E-16

 ITCOUNT  STATUS     NINF     OBJ VALUE        VECIN     VECOUT     NNEGDJ           DJ              NETA   NELEM
     1      I          1       0.22100002     GOOD2     GOOD1          2        -1.00000000            4      12
     1      O          0      -0.61999995     GOOD2     GOOD1          0         0.25000000            5      15
  MAS-COL
```

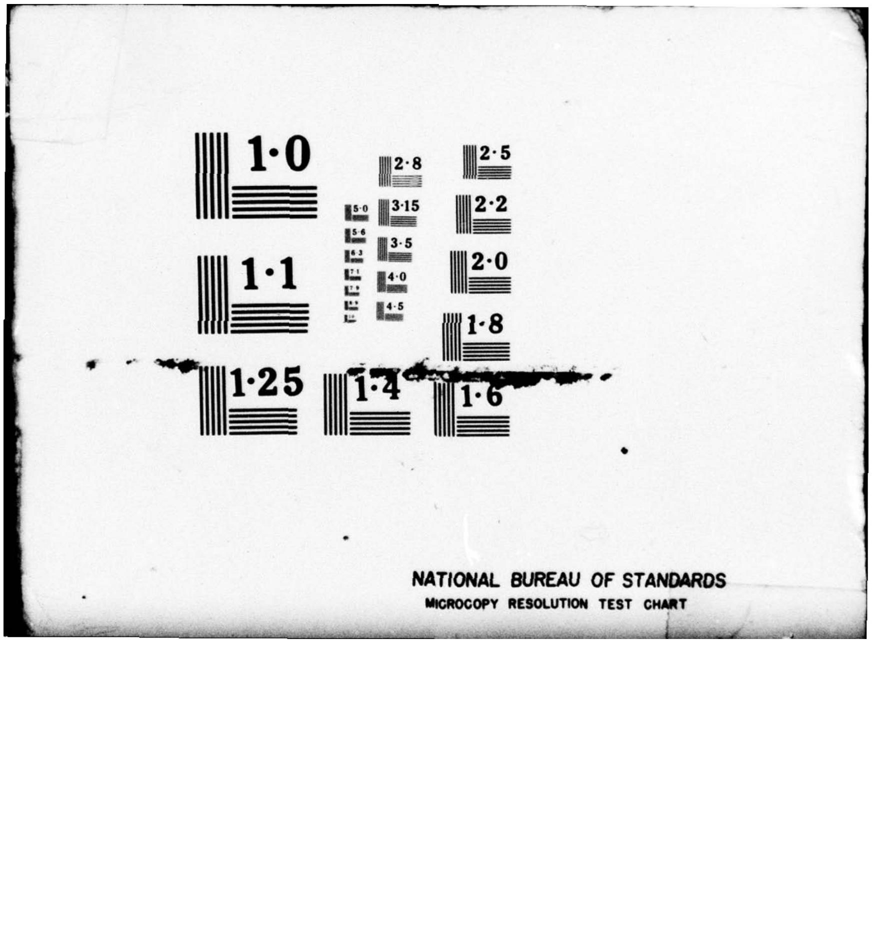
```

JH(I)                   VALUE          ROW NAMES        P(I)             PHS
ACT1                    0.90000000      UTIL1           0.50000000      -0.90000000
ACT2                    0.95000000      UTIL2           1.00000000      -0.95000000
ACT3                    3.92000000      UTIL3           0.25000000      -3.92000000
GOOD2                   0.22100005      GOOD1           1.00000000       3.00000000
MEXP                    0.62000000      GOOD2           0.0              3.00000000
OBJ                    -0.62000000      OBJ            -1.00000000       0.0
 THE TIME LEFT IS NOW 14.906462 SEC.





     STRUCTURAL COLUMNS OF MATRIX.





COLUMNS     1 THROUGH     3
   -0.9000      0.0       0.0
    0.0        -0.9500    0.0
    0.0         0.0      -3.9200
    1.0000      1.0000    1.0000
    1.0000      1.0000    1.0000
    0.0         0.0       0.0

 SMAL=  0.9090909 THETA=  0.4761905
 WE RETRACT ONTO THE    1 TH BOUNDARY.
X0=      0.000000    0.954545    0.231818
 INITIAL BUDGET SURPLUSES :
    0.550000      0.050000      0.020000

-----------------------------------------------
THE     1TH ENDPNT CALL:
THE CURRENT QUADRATIC APPROXIMATION TO THE CURVE IS:
     F(X)                    A2(*U)+              A3
         0.5500000          0.9718479          0.0
         0.0500000          0.0441749          0.0
         0.0200002          0.0706806          0.0
         0.0                0.2203737          0.4761905
 WE MOVE A DISTANCE         0.480119 ALONG THE APPROX.
THE CURVE HIT A  1 TYPE CONSTRAINT, NUMBER  4

 MFLAG=     2
ITERATION    0 X=     0.46660    0.02121    0.03394    0.58200
NORM(G(X))=    0.024803  G=    -0.024685   -0.002244   -0.000898    0.000000
ITERATION    1 X=     0.46660    0.02121    0.03394    0.61222
NORM(G(X))=    0.000000  G=    0.0         -0.000000    0.0          0.0
 AFTER   1 NEWTON ITERATIONS.
 THE    5TH PRIMAL VARIABLE WENT TO ZERO.
 VECIN: UTIL1      VECOUT: GOOD2

-----------------------------------------------
THE     2TH ENDPNT CALL:
THE CURRENT QUADRATIC APPROXIMATION TO THE CURVE IS:
     F(X)                    A2(*U)+              A3
         0.3166986          0.2583397          0.5000000
         0.0287908          0.4862849          0.0212092
```

```
          0.0115164         0.5888394         0.0339351
          0.0               0.5916551         0.6122210
 WE MOVE A DISTANCE          0.394011 ALONG THE APPROX.
THE CURVE HIT A  1 TYPE CONSTRAINT, NUMBER  5

 MFLAG=     2
 ITERATION :      0 X=     0.60179   0.21281   0.26594
NORM(F(X))=     0.065775 F=   0.048328 -0.044450 -0.003878
 ITERATION :      1 X=     0.63386   0.20485   0.30576
NORM(F(X))=     0.000313 F=   0.000128 -0.000255  0.000128
 ITERATION :      2 X=     0.63397   0.20470   0.30650
NORM(F(X))=     0.000000 F=   0.000000 -0.000000  0.000000
 ITERATION :      3 X=     0.63397   0.20470   0.30650
NORM(F(X))=     0.000000 F=   0.000000 -0.000000  0.000000
 AFTER    3 NEWTCN ITERATICNS.


AFTER    2 ENDPNT CALLS, WE HAVE ECUILIBRIUM.

TOTALS:  SCALAR FUNCTICN CALLS=       29
           JACOBIAN EVALUATICNS=       7

 THE TIME LEFT IS NOW 14.876805 SEC.
 UTILITY LEVELS:        1.5773504      1.1547005      4.2264982
 PRICES:      0.633975     0.866025     0.236603
 MAS-COL


JH(I)                   VALUE          ROW NAMES           PI(I)           RHS
ACT1                     1.57735041      UTIL1             0.63397455     -0.90000000
ACT2                     1.15470049      UTIL2             0.86602545     -0.95000000
ACT3                     4.22649816      UTIL3             0.23660254     -3.92000000
UTIL1                    0.67735041      GOOD1             0.73205090      3.00000000
MEXP                    -0.00000023      GOOD2             0.26794910      3.00000000
OBJ                      0.00000023      OBJ              -1.00000000      0.0
```

AD-A061 380   STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB   F/G 5/3
BCA AND HRA: TWO PROGRAMS FOR COMPUTING ECONOMIC EQUILIBRIA.(U)
AUG 78   T ELKEN   N00014-75-C-0267
UNCLASSIFIED   SOL-78-17   NL

2 OF 2
ADA
061380

END
DATE
FILMED
2-79
DDC


1·0
2·8
2·5
5·0
3·15
2·2
5·6
6·3
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

```
             IEF375I  JCB /TREW8668/ START 77201.1259        STOP  77201.1259 CPU    0MIN 00.19SEC
 &PARM1
 TOLFZ=   .1000000000000000000D-09,TCLBD=  .1000000000000000001D-03,TOLCV=  .9999999999999999997D-05,TOLEO=  .9999
550-06,
 ICNTRL=          1,IFCHC=            C,STPMX=  10.0000000000000000      ,STPRD=  .5000000000000000      ,ICECHC=
H=
          4,L=         3,IFFCC=          C,KOUTB=            0,ITLIME=          100
 &END
  IECHO,IH,ICECHO   0   4   1
 &PARAM
 ZTOLZE=  .100000005E-03,ZTOLPV=  .100000034E-05,ZTCOST=  .100000005E-03,ZSCALE= -1.00000000      ,NRMAX=
     2000,
 NFMAX=       8000,IALG=          C,NDEGI=            1,NDUAL=           999,NIPIW=        99999,KINP=              5,IFR
 0,IFCRSH=
          1,ICBJ=          8,ITCH=           30,INVFRQ=           30,ITRLIM=        99999,ZTOLRP=  .999999975E-05,Z
0000016E-01,
 ZTETA=  .100000043E-11,KINB=            C,ZTOLDA=  .100000008E-08
 &END
 PROBLEM  WHISMAN                                                                                    999999999999

                                                                                                          1,I
 PROBLEM STATISTICS
     8 ROWS
    11 STRUCTURAL COLUMNS
    36 NON-ZERO ELEMENTS
 DENSITY = 0.40909
   THE TIME LEFT IS NOW 14.919267 SEC.                                                                 350,NTMAX=
 INVERT STATISTICS
   20 NONZ IN BASIS                                                                                    AS=
     5 STRUCTURAL COLUMNS IN BASIS
     2 VECTORS ABOVE BUMP                                                                              TOLRI=   .100
     6 VECTORS BELOW BUMP
 L:      8 NONZ     2 ETAS
 U:      9 NONZ     3 ETAS
 TOTALS:    12 OFF DIAG NONZ     5 ETAS
 RELATIVE ERROR IN X =      0.0

 ITCOUNT  STATUS    NINF    OBJ VALUE      VECIN     VECOUT    NNEGDJ          DJ           NETA   NELEM
     1      I         2      9.95999908    C3AC3     MEXP         5       -6.00000000         5      17
     2      I         1      1.38999939    C3AC4     GOOD2        4       -2.50000000         6      22
     3      F         0     -0.38874996    MEXP      GOOD1        1       -1.00000000         7      26
     4      F         0     -0.94874996    C2AC2     C2AC1        3       -1.74999905         8      31
     5      F         0     -2.56499958    GOOD3     C3AC3        2       -0.37500000         9      37
     6      F         0     -2.57999992    C4AC2     C3AC4        1       -0.49999994        10      42
     6      0         0     -2.57999992    C4AC2     C3AC4        0        0.50000000        11      48
  WHISMAN

 JH(I)              VALUE        ROW NAMES        PI(I)           RHS
 C1AC1            0.48000000     UTIL1         1.50000000    -0.48000000
 C2AC2            0.32000000     UTIL2         2.50000000    -0.32000000
 C3AC1            5.90000000     UTIL3         0.50000000    -5.90000000
 C4AC1            0.92000000     UTIL4         1.00000000    -0.95000000
```

```
MEXP                   2.58000000     GOOD1              0.50000000     5.00000000
C4AC2                  0.03000000     GOOD2              0.50000000    11.00000000
GOOD3                  4.34000000     GOOD3             -0.00000000     9.00000000
OBJ                   -2.58000000     OBJ               -1.00000000     0.0
 THE TIME LEFT IS NOW 14.886756 SEC.




      STRUCTURAL COLUMNS OF MATRIX.






COLUMNS     1 THROUGH     4
   -0.4800     0.0        0.0        0.0
    0.0       -0.3200     0.0        0.0
    0.0        0.0       -5.9000     0.0
    0.0        0.0        0.0       -0.9500
    2.0000     1.0000     1.0000     1.0000
    2.0000     3.0000     5.0000     1.0000
    1.0000     3.0000     4.0000     1.0000
    0.0        0.0        0.0        0.0

 SMAL=   1.1718749 THETA=   0.5395683
 WE RETRACT ONTO THE    1 TH BOUNDARY.
XO=      0.000000     1.093750     0.441406     0.941406
 INITIAL BUDGET SURPLUSES :
     1.280000      1.200000      0.050000      0.050000

-----------------------------------------------------
THE     1TH ENDPNT CALL:
THE CURRENT QUADRATIC APPROXIMATION TO THE CURVE IS:
     F(X)                  A2(*U)+                A3
          1.2800001          0.8396748          0.0
          1.2000000          0.4723171          0.0
          0.0500002          0.0983998          0.0
          0.0500000          0.0491997          0.0
          0.0                0.2444579          0.5395683
 WE MOVE A DISTANCE           1.016266 ALONG THE APPROX.
THE CURVE HIT A  1 TYPE CONSTRAINT, NUMBER  5

 MFLAG=   2
 ITERATION :      0 X=     0.85333   0.48000   0.10000   0.05000
NORM(F(X))=     0.000000 F=    0.000000  0.000000  0.000000  0.000000
 ITERATION :      1 X=     0.85333   0.48000   0.10000   0.05000
NORM(F(X))=     0.0       F=    0.0       0.0       0.0       0.0
 AFTER    1 NEWTON ITERATIONS.


AFTER    1 ENDPNT CALLS, WE HAVE EQUILIBRIUM.

TOTALS:   SCALAR FUNCTION CALLS=      16
            JACOBIAN EVALUATIONS=      2

 THE TIME LEFT IS NOW 14.865749 SEC.
 UTILITY LEVELS:          1.3333334      0.8000000      6.0000004      1.0000000
 PRICES:       1.500000      2.500000      0.500000      1.000000
```

```
1109.     WHISMAN
1110.
1111.
1112.    JH(I)                  VALUE        ROW NAMES            PI(I)          RHS
1113.    C1AC1                  1.33333337     UTIL1             1.50000000    -0.48000000
1114.    C2AC2                  C.80000001     UTIL2             2.50000000    -0.32000000
1115.    C3AC1                  6.00000038     UTIL3             0.50000000    -5.90000000
1116.    C4AC1                  C.73333314     UTIL4             1.00000000    -0.95000000
1117.    MEXP                  -0.00000028     GOOD1             0.50000000     5.00000000
1118.    C4AC2                  0.26666687     GOOD2             0.50000000    11.00000000
1119.    GOOD3                  3.40000015     GOOD3            -0.00000000     9.00000000
1120.    OBJ                    0.00000028     OBJ              -1.00000000     0.0
1495.
1496.
1497.
```

III.6. HRA Source Listing

Below is a listing of HRA without the subroutines which comprise the linear programming code LPM1.

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 ZTOLZE,ZTOLPV,ZSCALE,ZTETA,ZTOLRP,ZTOLRI
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P,QN
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
     1A,S,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDEGI,NDUAL,NIPIW,IFEAS,IFCRSH
      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
      COMMON IA(4000),IE(8000)
C
      DATA QN/'N   '/
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/UNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/SCAL/ ST,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      COMMON/INT/ IPS(30),KDET,KOUNT,ISING,KEND
      COMMON/TOLER/ TOLFZ,TOLBD,TOLCV,THETA,STPMX,STPRD
      DIMENSION VS(10),F(10),SHR(10)
      NAMELIST/PARM1/TOLFZ,TOLBD,TOLCV,ICNTRL,IECHO,
     1STPMX,STPRD,ICECHO,IH,L,IPROD,KOUTE,ITLIME
C
C     HRA : A CODE WHICH IMPLEMENTS THE HOMOTOPY RETRACTION
C   ALGORITHM FOR SOLVING ECONOMIC EQUILIBRIUM PROBLEMS.
C
C     AUTHOR:    THOMAS R. ELKEN
C                SYSTEMS OPTIMIZATION LABORATORY
C,               OPERATIONS RESEARCH DEPARTMENT
C                STANFORD UNIVERSITY
C
C         FOR DESCRIPTION OF THE ALGORITHM AND DOCUMENTATION
C    OF THIS PROGRAM SEE SOL TECHNICAL REPORT 77-26.
C
      TOLBD= 1D-04
      TOLCV= 1D-05
      TOLFZ= 1D-10
      STPMX= 10.0
      STPRD= 0.5
      ITLIME= 100
      ICNTRL= 1
      IECHO= 0
      KOUTB= 0
      ICECHO= 0
      KFUN= 0
      KJAC= 0
      IPROD= 1
      READ (5,PARM1)
      WRITE (6,PARM1)
      IF (ICNTRL) 10,40,40
   10 CALL  INPUT(IACT)
      IF (IACT)  30, 20, 20
   20 IF (IECHO.EQ.0) GO TO 29
      MUB= NROW
      WRITE (6,21)
   21 FORMAT(/////,'             STRUCTURAL COLUMNS OF MATRIX.')
   22 LB= MUB+1
      MUB= LB+ 14
      K= 0
      DO 25 J= LB,MUB
          IF (J.GT.NCOL) GO TO 26
          K= K+ 1
          CALL UNPACK(J)
          DO 24  I= 1,NROW
              G1(I,K)= Y(I)
   24     CONTINUE
   25 CONTINUE
   26 WRITE (6,27)  LB,MUB
   27 FORMAT (/////,' COLUMNS ',I4,' THRUOGH ',I4)
      DO 23 I=1,NROW
      WRITE (6,28)   (G1(I,J),J=1,K)
   28 FORMAT (1X,15F8.3 )
   23 CONTINUE
      IF (MUB .LT. NCOL) GO TO 22
   29 CALL NORMAL
      CALL UNRAVL(0)
      GO TO 10
   30 STOP
   40 CALL  INPUT(IACT)
      CALL DEBUG(1)
      MFLAG= 0
      ITRY= 0
      TI= 5.
  343 IF (ITRY .LE. 10) GO TO 344
      WRITE (6,342)
  342 FORMAT('  MUST TERMINATE: TOO MANY TRIES AT INITIALIZATION
     1',/,'  OF V WITHOUT SUCCESS.')
  344 ITRY= ITRY + 1
      CALL NORMAL
      CALL UNRAVL(MFLAG)
      MFLAG= 0
      TI= TI+ 1.
      IF (MSTAT .NE. ON) GO TO 348
      DO 345  I= 1,IH
  345     B(I)= (1.0 -1./TI) * B(I)
      GO TO 343
  348 DO 349 I= 1,IH
          IF (Y(I) .GT. TOLBD) GO TO 349
          B(I)= (1.0 + 1./TI) * B(I)
          MFLAG= 1
  349 CONTINUE
      IF (MFLAG .GT. 0) GO TO 343
C
      CALL DEBUG(1)
      CALL SHIFTR(3,4)
      KEND= 0
      KENDSV= 0
      IHL= IH+ L
      M= NROW
      N= NCOL- NROW
      NM= N+ M
      L1= M- L
      MM1= M- 1
      ITSINV= 1
      IF (IPROD.EQ.0) GO TO 43
      READ (5.1030) (SHR(I).I=1.IH)
   43 ICOU=0
      ISAV= 0
      LC(1)= 1
      DO 68 K= 1.IH
         KP1= K+ 1
   45    READ(KINP.1010) ISHR. LL. KK
          IF (ISHR.EQ.1) GO TO 50
          IF (ISHR.EQ.-1) GO TO 42
          IF (LL.EQ.0) GO TO 58
          IF (KK.NE.0) GO TO 47
             ICOU= ICOU+ 1
             IC(ICOU)= LL
             C(ICOU)= B(LL)
             GO TO 45
   47     DO 48 I= LL. KK
             ICOU= ICOU+ 1
             IC(ICOU)= I
             C(ICOU)= B(I)
   48     CONTINUE
          GO TO 45
   42    IF (KK.NE.0) GO TO 44
            ICOU= ICOU+ 1
            IC(ICOU)= LL
            READ (5.1030) C(ICOU)
            GO TO 45
   44    ISAV= ICOU+ 1
         DO 46 I= LL.KK
            ICOU= ICOU+ 1
            IC(ICOU)= I
   46     CONTINUE
         READ(5.1030) (C(J).J=ISAV.ICOU)
         GO TO 45
   50     IF (KK.NE.0) GO TO 53
            ICOU= ICOU+ 1
            IC(ICOU)= LL
            C(ICOU)= SHR(K)*B(LL)
            GO TO 45
   53     DO 55 I= LL. KK
            ICOU= ICOU+ 1
            C(ICOU)= SHR(K)*B(I)
            IC(ICOU)= I
   55    CONTINUE
         GO TO 45
   58    LC(KP1)= ICOU+1
   68 CONTINUE
      IF (ICECHO.EQ.0) GO TO 69
      MUB= 0
      WRITE (6.61)
   61 FORMAT(/////.'            STRUCTURAL COLUMNS OF MATRIX.')
   62 LB= MUB+1
      MUB= LB+ IH - 1
      K= 0
      DO 65 J= LB.MUB
```

```
          IF (J.GT.NCOL) GO TO 66
          K= K+ 1
          CALL UPAKC(J)
          DO 64  I= 1,NROW
            G1(I,K)= Y(I)
   64     CONTINUE
   65 CONTINUE
   66 WRITE (6,67)  LB,MUB
   67 FORMAT (/////,' COLUMNS ',I4,' THROUGH ',I4)
      DO 74 I= 1,NROW
      WRITE (6,73) (G1(I,J),J=1,IH)
   73 FORMAT (1X,10F10.4)
   74 CONTINUE
   69 DO 70 I= 1,M
      IR= JH(I)
      XX(IR)= X(I)
      PI(I)= YTEMP(I)
      BA(I)= X(I)
   70 CONTINUE
      MP1= M+ 1
      DO 100 J= MP1,NM
      IF (KINBAS(J).NE. 0) GO TO 90
      DSUM= 0.
      LL= LA(J)
      KK= LA(J+1) - 1
      DO 80 I= LL,KK
      IR= IA(I)
      DE= A(I)
      DPROD= DE*YTEMP(IR)
      DSUM= DSUM + DPROD
   80 CONTINUE
      PI(J)= DSUM
      GO TO 100
   90 PI(J)= 0.
  100 CONTINUE
      K= 0
      DO 110 I= 1,NM
      IF (KINBAS(I) .EQ. 0) GO TO 105
      IDBAS(I)= 0
      GO TO 110
  105 K= K+ 1
      SIGMA(K)= I
      BB(K)= PI(I)
      IDBAS(I)= K
      XX(I)= 0.0
  110 CONTINUE
C        THIS IS THE BILINEAR PHASE OF THE ALGORITHM. VARIABLES XX(1).....
C  XX(M) ARE THE SLACKS. AND PI(1).....PI(M) ARE THE USUAL PI'S.
C  XX(M+1).....XX(M+N) ARE THE X'S. PI(M+1).....PI(M+N) ARE THE
C  DUAL SLACKS.  MORE INITIALIZATIONS.
C
      LP= 2
      ITAIL= 0
      PD= 1
      JS= 1
      DO 120 I=1,IH
      MUH(I)= 0
      NUH(I)= 0
      NU(I)= 0
      MU(I)= 0
```

```
120 CONTINUE
    KNU= 0
    KMU= 0
    DO 430  I= 1,IH
          IF (KINBAS(I).GT.0) GO TO 425
             CALL  SUPERB(1,1,I,0,0)
             GO TO 430
425       CALL  SUPERB(1,-1,I,0,0)
430 CONTINUE
    IF (KMU.EQ.0) GO TO 435
    DO 433  I= 1,KMU
433 VS(I)= PI(MU(I))
435 IF (KNU.EQ.0) GO TO 440
    DO 438  I= 1,KNU
          IK= I+ KMU
438       VS(IK)= XX(NU(I))
    CALL  BLCONS
440 CALL  FTN(F,VS)
    ID= 1
    THETA= 0.0
    WRITE (6,449) (F(I),I=1,IH)
449 FORMAT (1X,' INITIAL BUDGET SURPLUSES :',/,9F13.6)
    DO 130  I= 1,IH
          IF (F(I).GT.0) GO TO 130
          B(I)= B(I) + F(I)/PI(I)
          MFLAG= 1
130 CONTINUE
    IF (MFLAG .GT. 0) GO TO 343
    ZFLAG= 0
    GO TO 460
450 NTEMP(1)= NTEMP(1)+ JTINV
    CALL  INVERT
    CALL  RECALC
    ITSINV= 0
460 KEND= KEND+ 1
    ITSINV= ITSINV + 1
    IF (KEND .GT. ITLIME) STOP
    WRITE (6,222) KEND
222 FORMAT (/,' -------------------------------------------------------'
  1,/,' THE ',I4,'TH ENDPNT CALL:')
    CALL  ENDPNT(JS,PD1,IS,ZFLAG,NET)
    IF ( (KEND/3)*3 .NE. KEND ) GO TO 468
    CALL  DEBUG(4)
    CALL  DEBUG(1)
468 WRITE (6,470) NET
470 FORMAT (1X,' AFTER ',I3,' NEWTON ITERATIONS.')
    IF (ZFLAG.EQ.2) GO TO 920
    IF (PD1.EQ.1  .AND. IS.EQ.NM) GO TO 920
    IF (PD1) 300,310,310
300 WRITE (6,305) IS
305 FORMAT (1X,' THE ',I3,'TH DUAL VARIABLE WENT TO ZERO.')
    GO TO 480
310 WRITE (6,315) IS
315 FORMAT (1X,' THE ',I3,'TH PRIMAL VARIABLE WENT TO ZERO.')
480 IF (PD1.EQ.-1) GO TO 520
C
C    A PRIMAL VARIABLE WENT TO ZERO.  THAT VARIABLE MUST GO TO
C    NONBASIC, AND THE INCOMING VARIABLE IS DETERMINED BY FINDING THE
C    LARGEST ELEMENT IN THE APPROPRIATE ROW OF G1.
C
```

```
      PD= -1
      IROWP= KINBAS(IS)
      CALL FINDP(1,IS,JCOLP)
      INNAM1= ICNAM(JCOLP,1)
      INNAM2= ICNAM(JCOLP,2)
      IONAM1= ICNAM(IS,1)
      IONAM2= ICNAM(IS,2)
      WRITE(6,485) INNAM1,INNAM2,IONAM1,IONAM2
  485 FORMAT(1X,' VECIN: ',2A4,'  VECOUT: ',2A4)
      CALL PIVOT(JCOLP,IS)
      CALL UNPACK(JCOLP)
      CALL FTRAN(1)
      CALL WRETA
      CALL BSCNG(JCOLP,IS)
      CALL SUPERB(0,-1,JCOLP,1,JCOLP)
      CALL BLCONS
      JS= IS
      IF (ITSINV .GE. INVFRQ) GO TO 450
      GO TO 460
C
C        A DUAL VARIABLE WENT TO ZERO.  THAT VARIABLE MUST ENTER THE
C     BASIS, AND THE LEAVING VARIABLE MUST BE DETERMINED BY FINDING
C     THE LARGEST PIVOT ELEMENT IN THE IDBAS(IS)-ROW OF G2.
C
  520 PD= 1
      JCOLP= IS
      CALL FINDP(-1,IS,P)
      IROWP= KINBAS(P)
      INNAM1= ICNAM(IS,1)
      INNAM2= ICNAM(IS,2)
      IONAM1= ICNAM(P,1)
      IONAM2= ICNAM(P,2)
      WRITE (6,485) INNAM1,INNAM2,IONAM1,IONAM2
      CALL PIVOT(JCOLP,P)
      CALL UNPACK(JCOLP)
      CALL FTRAN(1)
      CALL WRETA
      CALL BSCNG(JCOLP,P)
      CALL SUPERB(0,1,P,-1,P)
      CALL BLCONS
      JS= IS
      IF (ITSINV.GE.INVFRQ) GO TO 450
      GO TO 460
C
C        ANOTHER BILINEAR CONSTRAINT IS SATISFIED.  WE INCREASE IH
C    AND WE HAVE A POINT IN W(IH).
C
  920 WRITE (6,930) KEND
  930 FORMAT(//,' AFTER',I4,' ENCPNT CALLS, WE HAVE EQUILIBRIUM.')
      WRITE (6,320) KFUN,KJAC
  320 FORMAT(/,' TOTALS:  SCALAR FUNCTION CALLS=',I8,/,11X,'JACOBIAN EVA
     1LUATIONS=',I8,/)
      CALL DEBUG(1)
      DO 935 I= 1,IH
  935 VS(I)= -B(I)+ XX(I)
      WRITE (6,940) (VS(I),I=1,IH)
      WRITE (6,945) (PI(I),I= 1, IH )
  940 FORMAT (1X,' UTILITY LEVELS: ',6F15.7)
  945 FORMAT (1X,'DUAL MULTPLIERS: ',9F12.6)
      DO 950 I= 1,NROW
```

```
         IV= JH(I)
         X(I)= XX(IV)
         Y(I)= PI(I)
  950 CONTINUE
      CALL UNRAVL(0)
      STOP
 1010 FORMAT (10I4)
 1030 FORMAT (10F7.4 )
      END
      SUBROUTINE BLCONS
C
C       WE ARE GOING TO CALCULATE THE IH ROWS OF COEFFICIENTS
C    FOR THE BILINEAR EQUATIONS AND THE BILINEAR INEQUALITY.
C      THE BASIC MATHEMATICAL STRUCTURE IS THE FOLLOWING:
C
C         BF1 + D1*PI(MU)- DIAG(PI(MU))*(F1*X(NU) + E1) = 0
C         BF2 + D2*PI(MU)- DIAG(X(NU))*(F2*PI(MU) + E2) = 0
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
     1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDEGI,NDUAL,NIPIW,IFEAS,IFCRSH
      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
      COMMON IA(4000),IE(8000)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      K1= 0
      K2= 0
      IF (KMU.EQ.0) GO TO 100
      DO 80 K= 1,KMU
      K1= MU(K)
      IF (K1.GT.IH) GO TO 80
      IDD= KINBAS(K1)
      IF (KNU.EQ.0) GO TO 45
      DO 40 I= 1,KNU
   40 F1(K,I)= G1(IDD,I)
   45 E1(K)= BA(IDD)
      CALL UPAKC(K1)
      DO 60 I= 1,KMU
      DSUM  = 0.0
      IF (MU(I) .LE. M)   DSUM= Y(MU(I))
      DO 50 J= 1,M
         IDJ= IDBAS(J)
         IF (IDJ .EQ. 0) GO TO 50
```

```
            DSUM= DSUM+ Y(J)*G2(IDJ,I)
   50 CONTINUE
      D1(K,I)= DSUM
   60 CONTINUE
      BF1(K)= 0.
      DO 70 J= 1,M
      IF(IDBAS(J).EQ.0) GO TO 70
      BF1(K)= BF1(K)+     Y(J)*BB(IDBAS(J))
   70 CONTINUE
   80 CONTINUE
C
C     NOW THE SECOND TYPE OF EQUATION IS CALCULATED BECAUSE
C   PI(K) IS BASIC.
C
  100 IF (KNU.EQ.0) RETURN
      DO 145 K= 1,KNU
      K2= NU(K)
      IF(K2.GT.IH) GO TO 145
      IDD= IDBAS(K2)
      IF (KMU.EQ.0) GO TO 123
      DO 120 I= 1,KMU
  120 F2(K,I)= G2(IDD,I)
  123 E2(K)= BB(IDD)
      CALL UPAKC(K2)
      IF (KMU.EQ. 0) GO TO 132
      DO 130 I= 1,KMU
      D2(K,I)= 0.0
      IF (MU(I) .LE. M)   D2(K,I)= Y(MU(I))
      DO 125 J= 1,M
      IF( IDBAS(J).EQ.0) GO TO 125
      D2(K,I)= D2(K,I)+ Y(J)*G2(IDBAS(J),I)
  125 CONTINUE
  130 CONTINUE
  132 BF2(K)= 0.
      DO 140 J= 1,M
      IF (IDBAS(J).EQ.0) GO TO 140
      BF2(K)= BF2(K)+ Y(J)*BE(IDEAS(J))
  140 CONTINUE
      IF (K2.NE.IH) GO TO 145
  145 CONTINUE
      RETURN
      END
      SUBROUTINE FINDP(PD1,IS,P)
C
C     THIS SUBROUTINE CHOOSES THE VARIABLE TO BECOME IMPLICITLY
C   BASIC AS THE ONE WITH THE PIVOT ELEMENT LARGEST IN
C   ABSOLUTE VALUE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/  JH,DIGMA,KINEAS, IDBAS
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      COMMON/TOLER/ TOLFZ,TOLBD,TOLCV,THETA,STPMX,STPRD
      P= 0
      IF (PD1.EQ.-1) GO TO 30
      IDD= KINBAS(IS)
      DO 20 I= 1,KNU
      IF (P.NE.0) GO TO 10
      BIG=  DABS(G1(IDD,I))
      P= NU(I)
      GO TO 20
   10 COMP= DABS(G1(IDD,I))
      IF (COMP.LE.BIG) GO TO 20
      BIG= COMP
      P= NU(I)
   20 CONTINUE
      IF (BIG.LT.TOLFZ) P= 0
      RETURN
   30 IDD= IDBAS(IS)
      DO 40 J= 1,KMU
      IF (P.NE.0) GO TO 35
      BIG= DABS(G2(IDD,J))
      P= MU(J)
   35 COMP= DABS(G2(IDD,J))
      IF (COMP.LE.BIG) GO TO 40
      BIG= COMP
      P= MU(J)
   40 CONTINUE
      IF (BIG.LT.TOLFZ) P= 0
      RETURN
      END
      SUBROUTINE PIVOT (SS,RR)
C
C       THIS ROUTINE PERFORMS A PIVOT ON THE PRIMAL SUPER-
C   BASIC COLUMNS IF PD.EQ.1. DUAL IF PD.EQ.-1. THE PIVOT
C   BRINGS COLUMN SS INTO THE BASIS AND COLUMN RR OUT OF THE
C   BASIS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
       COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
     1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),FUN(8),
     6IPUNC,NDEGI,NDUAL,NIPIW,IFBAS,IFCRSH
       COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
       COMMON IA(4000),IE(8000)
       COMMON/LP1/PI(1302),XX(1302)
       COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
       COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
       COMMON/INDX2/ JH,DIGMA,KINBAS, IDBAS
       COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
       EPS= 16.**(-13)
       RB= KINBAS(RR)
       IF (SS .GT. IH) GO TO 5
       IF (NUH(SS).NE.0) GO TO 20
     5 CALL UNPACK(SS)
       CALL FTRAN(1)
       YTEMP(RB)= -1./Y(RB)
       DO 10 I= 1,M
```

```
          IF (I.NE.RB) YTEMP(I)= Y(I)*YTEMP(RB)
       10 CONTINUE
          GO TO 30
       20 JS= NUH(SS)
          YTEMP(RB)= 1./G1(RB,JS)
          DO 25 I= 1.M
          IF (I.NE.RB) YTEMP(I)= -G1(I.JS)*YTEMP(RB)
       25 CONTINUE
       30 IF (KNJ.EQ.0) GO TO 55
          DO 50 J= 1,KNU
          IF (NU(J).EQ.SS) GO TO 50
          V= G1(RB,J)
          G1(RB,J)= 0.
          DO 40 I= 1,M
       40 G1(I.J)= G1(I.J)+ V*YTEMP(I)
          G1(RB,J)= -G1(RB,J)
       50 CONTINUE
       55 V= BA(RB)
          BA(RB)= 0.
          DO 60 I= 1.M
       60 BA(I)= BA(I)+ V*YTEMP(I)
          BA(RB)= -BA(RB)
C
C            WE ARE NOW PIVOTING ON THE DUAL SYSTEM.  IF THIS
C       IS NOT AN IMPLICIT BASIC-TYPE PIVOT. WE CALL SUPERB
C       TO CALCULATE THE PIVOT COLUMN.
C
          IFL= 0
          RB= IDBAS(SS)
          IF (RR .GT. IH) GO TO 70
          IF (MUH(RR).NE.0) GO TO 80
       70 IFL= 1
          CALL SUPERB(1,-1,RR,0,0)
          JS= KMU
          GO TO 82
       80 JS= MUH(RR)
       82 YTEMP(RB)= 1./G2(RB.JS)
          DO 85 I= 1.N
          IF (I.NE.RB) YTEMP(I)= -G2(I.JS)*YTEMP(RB)
       85 CONTINUE
          IF (KMU.EQ.0) GO TO 115
          DO 110 J= 1,KMU
          IF (MU(J).EQ.RR) GO TO 110
          V= G2(RB.J)
          G2(RB,J)= 0.
          DO 100 I= 1.N
      100 G2(I,J)= G2(1.J)+ V*YTEMP(I)
          G2(RB,J)= -G2(RB.J)
      110 CONTINUE
      115 V= BB(RB)
          BB(RB)= 0.0
          DO 120 I= 1.N
      120 BB(I)= BB(I)+ V*YTEMP(I)
          BB(RB)= -BB(RB)
          IF (IFL.EQ.0) RETURN
          CALL SUPERB (2.0.0.-1.RR)
          RETURN
          END
C
C
```

```
      SUBROUTINE UPAKC(II)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DF,B(350),X(350),Y(350),YTEMP(350),
     1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDEGI,NDUAL,NIPIW,IFBAS,IFCRSH
      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
      COMMON IA(4000),IE(8000)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
C
      DO 100 I= 1,M
      Y(I)= 0.
  100 CONTINUE
C
      LL= LC(II)
      KK= LC(II+1) - 1
      DO 200 I= LL,KK
      IR= IC(I)
      Y(IR)= C(I)
  200 CONTINUE
C
      RETURN
      END
C
C        THE FOLLOWING ROUTINE MAKES THE CHANGES IN THE INDEX
C   SETS NECESSARY EVERY TIME A BASIS CHANGE IS MADE.
C
      SUBROUTINE BSCNG(S,R)
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      INTEGER  S,SSAV,R,RSAV
      RSAV= KINBAS(R)
      JH(RSAV)= S
      KINBAS(R)= 0
      KINBAS(S)= RSAV
      SSAV= IDBAS(S)
      DIGMA(SSAV)= R
      IDBAS(S)= 0
      IDBAS(R)= SSAV
      RETURN
      END
C       THIS SUBROUTINE CAN DO THREE THINGS DETERMINED BY THE
C   PARAMETER 'KEY'.  IT CAN ADD A COLUMN, REMOVE A COLUMN, OR
C   BOTH ADD AND REMOVE COLUMNS FROM THE PRIMAL OR DUAL SUPER-
C   BASIC COLUMNS, DEPENDING ON WHETHER KEY IS 1, 2, OR 0, RESP-
C   ECTIVELY.  PD1 IS 1 OR -1 DEPENDING ON WHETHER A PRIMAL OR
C   DUAL COLUMN IS BEING ADDED.  PD2 IS A SIMILAR FLAG FOR THE
C   COLUMN BEING REMOVED.  IS AND JS INDICATE THE PARTICULAR
C   COLUMN TO BE ADDED OR REMOVED FROM THE GROUP OF COLUMNS
C   SPECIFIED BY PD1 AND PD2.
C
```

```
      SUBROUTINE SUPERB(KEY,PD1,IS,PD2,JS)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
     1A,F,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDEGI,NDUAL,NIPIW,IFEAS,IFCRSH
      COMMON  ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
      COMMON  IA(4000),IE(8000)
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/ INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPC,KFUN,KJAC
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      EPS= 16.**(-13)
      IF (KEY.EQ.1) GO TO 50
      IF (PD2.GT.0) GO TO 20
      KM= KMU
      KMU= KMU- 1
      IF (JS .GT. IH) GO TO 4
      MH= MUH(JS)
      MUH(JS)= 0
      IF (MH.EQ.KM) GO TO 6
      DO 5 I= MH,KMU
      IM= MU(I+1)
      MU(I)= IM
       MUH(IM)= MUH(IM)- 1
      DO 5 J= 1,N
      G2(J,I)= G2(J,I+1)
    5 CONTINUE
    6 MU(KM)= 0
    4 DO 7 J= 1,N
    7 G2(J,KM)= 0.0
      IF (KEY.EQ.2) RETURN
      GO TO 50
   20 KN= KNU
      KNU= KNU- 1
      IF (JS .GT. IH) GO TO 28
      NH= NUH(JS)
      NUH(JS)= 0
      IF (NH.EQ.KN) GO TO 26
      DO 25 I= NH,KNU
      JN= NU(I+1)
      NU(I)= JN
      NUH(JN)= NUH(JN)- 1
      DO 25 J= 1,M
      G1(J,I)= G1(J,I+1)
   25 CONTINUE
   26 NU(KN)= 0
   28 DO 30 I= 1,M
   30 G1(I,KN)= 0.0
```

```
      IF (KEY.EQ.2) RETURN
C
C      NOW WE ADD THE SUPERBASIC VARIABLE SPECIFIED BY PD1.IS.
C
   50 IF (PD1.LT.0) GO TO 70
      KN= KNU
      KNU= KNU+ 1
      IF (IS.GT.IH) GO TO 55
      NU(KNU)= IS
      NUH(IS) = KNU
   55 CALL UNPACK(IS)
      CALL FTRAN(1)
      DO 60 I= 1,M
   60 G1(I,KNU)= -Y(I)
      RETURN
C
C        NOW WE ARE BRINGING A DUAL SUPERBASIC COLUMN IN.
C
   70 KMU= KMU+ 1
      IF (IS.GT.IH) GO TO 73
      MU(KMU)= IS
      MUH(IS)= KMU
   73 IF (IS.LE.M) GO TO 110
C
C      THE ENTERING VARIABLE IS A ZETA.  THE ENTERING COLUMN
C   IS A ROW OF INV(B) AND A ROW OF INV(B)*D.
C
      IDD= KINBAS(IS)
      DO 75 I= 1,M
   75 Y(I)= 0.
      Y(IDD)= 1.
      CALL BTRAN
      CALL SHIFTR(3,2)
      DO 80 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 80
      G2(IDBAS(I),KMU)= X(I)
   80 CONTINUE
      DO 100 J= MP1,NM
      IF (IDBAS(J).EQ.0) GO TO 100
      CALL UNPACK(J)
      DOT= 0.0
      DO 90 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 90
      DOT= DOT+ Y(I)*X(I)
   90 CONTINUE
      G2(IDBAS(J),KMU)= DOT
  100 CONTINUE
      RETURN
C
C        THE ENTERING VARIABLE IS A PI.  THE ENTERING COLUMN
C   IS PARTLY C1*INV(B) AND PARTLY C2+ C1*INV(B)*D.
C   -C1*INV(B) IS PART OF THE BASIS INVARSE CORRESPONDING TO
C   THE X-BASIC COLUMNS AND THE T-BASIC ROWS.
C
  110 IDD= KINBAS(IS)
      DO 130 I= 1,M
  130 Y(I)= 0.
      Y(IDD)= 1.
      CALL BTRAN
      CALL SHIFTR(3,2)
      DO 140 I= 1.M
      IF (IDBAS(I).EQ.0) GO TO 140
      G2(IDBAS(I).KMU)= X(I)
  140 CONTINUE
      DO 160 J= MP1.NM
      IF (IDBAS(J).EQ.0) GO TO 160
      CALL UNPACK(J)
      DOT= 0.0
      DO 150 I= 1.M
      IF (IDBAS(I).EQ.0) GO TC 150
      DOT= DOT+ Y(I)*X(I)
  150 CONTINUE
      G2(IDBAS(J).KMU)= Y(IS)+ DOT
  160 CONTINUE
      RETURN
      END
      SUBROUTINE RECALC
C
C         THIS SUBROUTINE RECALCULATES THE SUPERBASIC COLUMNS
C  USING THE CURRENT BASIS IN GR FORM.  IT CAN ALSO ADD
C  A VARIABLE AND COLUMN TO THE SUPERBASICS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINEAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,CCND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
     1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(350),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDEGI,NDUAL,NIPIW,IFBAS,IFCRSH
       COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
       COMMON IA(4000),IE(8000)
       COMMON/LP1/PI(1302),XX(1302)
       COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
       COMMON/INDX1/ NUH(10),MUH(10),NL(10),MU(10)
       COMMON/INDX2/ JH,DIGMA,KINEAS,IDBAS
       COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
       COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
       EPS= 16.**(-13)
       IF (KNU.EQ.0) GO TO 26
       DO 28 J= 1,KNU
       IDD= NU(J)
       CALL UNPACK(IDD)
       CALL FTRAN(1)
       DO 25 I= 1.M
    25 G1(I.J)= - Y(I)
    28 CONTINUF
    26 CALL SHIFTR(1.3)
       CALL FTRAN(1)
       DO 22 I= 1.M
    22 BA(I)= Y(I)
    30 IF (KMU.EQ.0) GO TO 140
       DO 130 J= 1.KMU
       IDD= MU(J)
      IF (IDD.LE.M) GO TO 70
C
C     ZETA- VARIABLE IS ENTERING
C
      IB= KINBAS(IDD)
      DO 35 I= 1,M
   35 Y(I)= 0.
      Y(IB)= 1.
      CALL BTRAN
      CALL SHIFTR(3,2)
      DO 40 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 40
      G2(IDBAS(I),J)= X(I)
   40 CONTINUE
      DO 60 K= MP1,NM
      IF (IDBAS(K).EQ.0) GO TO 60
      CALL UNPACK(K)
      DOT= 0.0
      DO 50 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 50
      DOT= DOT+ Y(I) * X(I)
   50 CONTINUE
      G2(IDBAS(K),J)= DOT
   60 CONTINUE
      GO TO 130
C
C     PI- VARIABLE IS ENTERING
C
   70 IB= KINBAS(IDD)
      DO 90 I= 1,M
   90 Y(I)= 0.
      Y(IB)= 1.
      CALL BTRAN
      CALL SHIFTR(3,2)
      DO 100 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 100
      G2(IDBAS(I),J)= X(I)
  100 CONTINUE
      DO 120 K= MP1,NM
      IF (IDBAS(K).EQ.0) GO TO 120
      CALL UNPACK(K)
      DOT= 0.0
      DO 110 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 110
      DOT= DOT+ Y(I) *X(I)
  110 CONTINUE
      G2(IDBAS(K),J)= Y(IDD)+ DOT
  120 CONTINUE
  130 CONTINUE
  140 DO 150 I= 1,M
  150 Y(I)= 0.0
      IF (KINBAS(NM).NE.0)  Y(KINBAS(NM))= 1.
      CALL BTRAN
      CALL SHIFTR(3,2)
      DO 160 I= 1,M
  160 IF (IDBAS(I).NE.0)  BB(IDBAS(I))= X(I)
      DO 170 J= MP1,NM
      IF (IDBAS(J).EQ.0) GO TO 170
      CALL UNPACK(J)
      DOT= 0.0
      DO 165 I= 1,M
      IF (IDBAS(I).EQ.0) GO TO 165
      DOT= DOT+ X(I)*Y(I)
  165 CONTINUE
      BB(IDBAS(J))= DOT
  170 CONTINUE
      RETURN
      END
      SUBROUTINE ENDPNT(JS,PD1,IS,ZFLAG,NET)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MIN,MIN2
      REAL*4 C(800)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      COMMON/INT/ IPS(30),KDET,KOUNT,ISING,KEND
      COMMON/TOLER/ TOLFZ,TOLEC,TOLCV,THETA,STPMX,STPRD
      DIMENSION U(3),V1(10),V2(10),F(10),DOT(4),RHS(10),UL(10,10)
      DIMENSION G(10)
C
C        THIS SUBROUTINE WILL FIND THE OPPOSITE ENDPOINT OF
C    G**(-1)(0) WITH RESPECT TO THE CELL DEFINED BY THE CURR-
C    ENT BASIS, WHERE
C             G(X,THETA)= THETA*F(X) - (1-THETA)*X
C             F(X)= THE BUDGET SURPLUSES DETERMINED BY THE
C                   FTN AND BLCONS SUBROUTINES.
C             THETA= THE HOMOTOPY PARAMETER.
C
      NET= 0
      INFL= 0
      FRAC=0.5
      DELTA = .01
      TWODEL= 2.0*DELTA
C
C        MFLAG=0 WHEN WE ARE SOLVING FOR THE QUADRATIC APPROXIMATION
C        MFLAG=1 WHEN WE ARE SOLVING ONE OF THE HYPERPLANE SUBPROBLEMS.
C        MFLAG=2 WHEN ONE OF THE LINEAR CONSTRAINTS IS BINDING.
C        MFLAG=3 WHEN THE EQUILIBRIUM POINT APPEARS TO BE IN THE CURRENT
C    CELL.  NEWTON'S METHOD WILL BE IMPLEMENTED AS A TAIL ROUTINE ON F.
C
      MFLAG=0
      KK= 0
      ITRY= 0
      IHP1= IH+ 1
      F(IHP1)= 0.0
      KMU1= KMU+ 1
      ICX= IHP1
C           INITIALIZE THE INDEPENDENT VARIABLES IN X.
      IF (KMU.EQ.0) GO TO 615
      DO 610 I= 1,KMU
  610 X(I)= PI(MU(I))
  615 IF (KNU.EQ.0) GO TO 625
      DO 620 I= 1.KNU
      KM= KMU+ I
  620 X(KM)= XX(NU(I))
      X(IHP1)= THETA
      ITER= 0
  625 ISGCT= 0
    5 ITER= ITER+ 1
      CALL GTN(O.G.X.F)
      CALL DERIVG(O,X.G.F)
    7 ISGCT= ISGCT+ 1
      IF (ISGCT.GT.IH) GO TO 300
      DO 10 I= 1.IH
          RHS(I)= -H(I.ICX)
   10 CONTINUE
      IF (ICX.EQ.IHP1) GO TO 35
      DO 24 J= ICX.IH
          JP1= J+ 1
          DO 23 I= 1.IH
   23         H(I.J)= H(I.JP1)
   24 CONTINUE
C
C        SOLVE THE LINEAR SYSTEM TO FIND THE TANGENT TO
C        THE CURVE DEFINED BY G**-1(0).
C
   35 CALL DECOMP(IH.H.UL)
   37 IF (ISING.NE.1) GO TO 38
          IF (ICX .EQ. 1) GO TO 36
          ICX= ICX- 1
          GO TO 7
   36     ICX= IH
          GO TO 7
   38 CALL SOLVE(IH.UL.RHS.V2)
   40 DO 39 I= 1.IH
          IM= IHP1- I
          IF (IM.LT.ICX) GO TO 41
              IP1= IM+ 1
              V2(IP1)= V2(IM)
   39 CONTINUE
   41 V2(ICX)= 1.0
      IF (ITER.GT.1) GO TO 45
      SUM= 0.0
      IF (PD.EQ.-1) GO TO 110
       JB= KINBAS(JS)
      IF (JB.NE.0) GO TO 104
              IF (V2(ID).GT.0) GO TO 135
              GO TO 140
  104 DO 105 I= 1.KNU
          IK= KMU+ I
          SUM= SUM+ V2(IK)*G1(JB.I)
  105 CONTINUE
      IF (SUM.LT.0.0) GO TO 140
      GO TO 135
  110 JB= IDBAS(JS)
      IF (JB.NE.0) GO TO 112
              IF (V2(ID).GT.0) GO TO 135
              GO TO 140
  112 DO 115 I= 1.KMU
          SUM= SUM+ V2(I)*G2(JB.I)
  115 CONTINUE
```

```
C
C          INIT IS THE CURVE INDEX; IT TELLS US WHICH DIRECTION
C      THE TANGENT MUST POINT TO KEEP GOING IN THE SAME DIRECTION.
C
      IF (SUM.LT. 0.0) GO TO 140
  135 INIT= KDET
      GO TO 45
  140 INIT= -KDET
   45 CALL NORM(V2,S1,IHP1)
      S1= (KDET*INIT)/S1
      DO 47 I= 1,IHP1
          ACC(2,I)= V2(I)*S1
          ACC(3,I)= X(I)
   47 CONTINUE
      WRITE (6,850)
      DO 48 J= 1,IHP1
          WRITE (6,855) F(J), (ACC(I,J),I=2,3)
   48 CONTINUE
C
C          FIND THE FIRST BOUNDARY THAT THE TANGENT, Q(ALPHA)
C     HITS.  WE ARE TRYING TO FIND
C     THE SMALLEST ALPHA SUCH THAT
C              G(I,.)*Q(ALPHA) + GB(I) = 0
C
      IFLAG= 0
      IF (KNU.EQ.0) GO TO 249
      DO 250 I= 1,M
      IV= JH(I)
      IF (IV.EQ.M .OR. IV.LE.IH) GO TO 250
          DO 240 LL= 2,3
              U(LL)= 0.0
              DO 230 K= 1,KNU
                  KJ= K+ KMU
                  U(LL)= U(LL)+ G1(I,K)*ACC(LL,KJ)
  230         CONTINUE
  240     CONTINUE
  241     U(3)= U(3)+ BA(I)
          IF (DABS(U(2)).LT.TOLFZ) GO TO 250
          ALPHA= -U(3)/U(2)
          IF (ALPHA .LT. -TOLFZ) GO TO 250
          IF (ALPHA .GT. TOLBC) GO TO 244
              TOD= 0.0
              DO 242 K= 1,KNU
                  KM= K+ KMU
  242             TOD= TOD+ G1(I,K)*ACC(2,KM)
              IF (TOD .GT. -TOLFZ) GO TO 250
  244     IF (IFLAG.EQ.1) GO TO 243
          IFLAG = 1
          JJ= I
          MIN= ALPHA
          MPD= 1
          GO TO 250
  243     IF (ALPHA.GE.MIN)  GO TO 250
          JJ= I
          MIN= ALPHA
          MPD= 1
  250 CONTINUE
  249 IF (KMU.EQ.0) GO TO 640
      DO 259 I= 1,N
          IF (SIGMA(I) .EQ. M) GO TO 259
```

```
           DO 256 LL= 2,3
               U(LL)= 0.0
               DO 254 K= 1,KMU
                    U(LL)= U(LL)+ G2(I,K)*ACC(LL,K)
  254          CONTINUE
  256      CONTINUE
  257      U(3)= U(3)+ BB(I)
           IF (DABS(U(2)).LT.TOLFZ) GO TO 259
           ALPHA= -U(3)/U(2)
           IF (ALPHA .LT. -TOLFZ) GO TO 259
           IF (ALPHA.GT. TOLBD) GO TO 258
               TOD= 0.0
               DO 252 K= 1,KMU
                    TOD= TOD+ ACC(2,K)*G2(I,K)
  252          CONTINUE
               IF (TOD .GT. -TOLFZ) GO TO 259
  258      IF (IFLAG.EQ.1) GO TO 253
           IFLAG = 1
           JJ= I
           MIN= ALPHA
           MPD= -1
           GO TO 259
  253      IF (ALPHA.GE.MIN)  GO TO 259
           JJ= I
           MIN= ALPHA
           MPD= -1
  259 CONTINUE
  540 MFLAG= 2
      IF (KEND.GT.1 .OR. INFL.EQ.1) GO TO 251
           MIN= MIN* 0.25
           MFLAG= 1
           INFL= 1
  635 FORMAT (1X,' WE MOVE A DISTANCE ',F15.6,' ALONG THE APPROX.',/,
     1' THE CURVE HIT A',I3,' TYPE CONSTRAINT, NUMBER',I3)
C
C     SET THE INITIAL X EQUAL TO Q(ALPHA) AS AN APPROX-
C     IMATION TO THE ENDPOINT SOLUTION.
C
  251 WRITE (6,635) MIN, MPD, JJ
      IF (MIN.LT. STPMX) GO TO 255
           MIN= MIN* STPRC
           MFLAG =  1
           WRITE (6,635) MIN, MPD, JJ
  255 BT= 0.
      DO 290 I=1,IHP1
           X(I)= MIN * ACC(2,I)  +  ACC(3,I)
           BT= BT + X(I)*ACC(2,I)
  290 CONTINUE
      WRITE (6,292) MFLAG
  292 FORMAT(/,1X,' MFLAG= ',I4)
      IF (MPD .EQ. -1) GO TO 293
      IF (JH(JJ) .EQ. NM) GO TO 295
  293 IF (ITAIL .EQ. 1) GO TO 295
  264 IT= 0
      ITRY= ITRY+ 1
      IF (ITRY .LE. 6) GO TO 265
           ITAIL= 1
           GO TO 269
  265 CALL GTN(MFLAG,G,X,F)
      CALL NORM(G,S1,IHP1)
```

```
        WRITE (6,935) IT, (X(I),I=1,IHP1)
        WRITE(6,936) S1,(G(J),J=1,IHP1)
        CALL DERIVG(MFLAG,X,G,F)
        IF (S1.LT.TOLCV) GO TO 280
        IT= IT+ 1
        NET= NET+ 1
        IF (IT.LE.10) GO TO 265
            MFLAG= 1
            MIN= MIN/2.
            IF (MIN .GT. TOLRD) GO TO 255
            STOP
    266 IF (NET .LT. 60) GO TO 268
    269 WRITE (6,267)
    267 FORMAT ('  ENDPOINT HAS FAILED --- TOO MANY ITERATIONS.')
        STOP
    268 CALL DECOMP(IHP1,H,UL)
        IF (ISING.EQ.1) GO TO 300
        CALL SOLVE(IHP1,UL,G,Z)
        DO 270 I= 1,IHP1
    270     X(I)= X(I)- Z(I)
        GO TO 265
    280 IF (X(IHP1) .GT. 1.0001) GO TO 295
        IF (MFLAG.EQ.2) GO TO 800
        GO TO 625
C
C           NEWTON'S METHOD FOR THE PROBLEM OF FINDING THE INTERSECTION
C     OF THE CURVE WITH THE CONSTRAINT DEFINED BY MPD AND JJ.
C
    295 MFLAG= 3
        DO 100 K= 1,21
            KM1= K- 1
            CALL FTN(F,X)
            CALL NORM(F,S1,IH)
        WRITE (6,938) KM1, (X(I),I=1,IH)
        WRITE (6,939) S1, (F(I),I=1,IH)
            IF (S1.LT.TOLFZ) GO TO 800
            CALL DERIV
            NET= NET+ 1
            IF (IH.GT.1) GO TO 55
                IF (DABS(H(1,1)) .LT. TOLFZ) GO TO 58
                Z(1)= F(1)/H(1,1)
                GO TO 65
     55     CALL DECOMP(IH,H,UL)
            IF (ISING.EQ.0) GO TO 60
     58     WRITE (6,940)
            STOP
     60     CALL SOLVE(IH,UL,F,Z)
     65     ALPH= 1.
            IF (ITAIL.EQ.0) GO TO 80
            CALL DSCNT(ALPH,S1)
            IF (ALPH.GT. TOLCV) GO TO 70
            WRITE (6,943)
            GO TO 300
     70     WRITE(6,955) ALPH
     80     DO 95 I= 1,IH
                X(I)= X(I)- ALPH*Z(I)
     95     CONTINUE
    100 CONTINUE
                GO TO 300
    300 CALL CONCHK(GMIN,KGMIN,MP2)
```

```
      THETA= X(IHP1)
      IF (KMU.EQ.0) GO TO 146
      DO 145 I= 1,KMU
         KM= MU(I)
145      PI(KM)= X(I)
146   IF (KNU.EQ.0) GO TO 148
      DO 147 I= KMU1,IH
         IK= I- KMU
         KM= NU(IK)
147      XX(KM)= X(I)
148   IF (GMIN .LT. -TOLBD) GO TO 170
      IF (MFLAG.LE.2) GO TO 153
      ZFLAG= 2
      RETURN
153   IF (MFLAG.EQ.1) GO TO 150
155   IF (MPD.EQ. -1) GO TO 151
      IS= JH(JJ)
      PD1= 1
      RETURN
151   IS= DIGMA(JJ)
      PD1= -1
      RETURN
150   MFLAG= 0
      GO TO 625
C
C          IF THE CURRENT VALUE OF X VIOLATES THE CONSTRAINT DEFINED
C     BY KGMIN AND MP2, FIND A GOOD STARTING POINT FOR NEWTON'S
C     METHOD BY FINDING THE POINT ON THE LINE SEGMENT
C                  ACC(1,.) + ALPHA*(X - ACC(1,.)),     0<= ALPHA<= 1
C     THAT SATISFIES THE VIOLATED CONSTRAINT EXACTLY.
C
170   JJ= KGMIN
      MPD= MP2
      MFLAG= 2
      WRITE (6,174) MPD,JJ,GMIN
174   FORMAT(/,' THE QUADRATIC PICKED THE WRONG CONSTRAINT.',/
     1,'  THE MOST INFEASIBLE CONSTRAINT IS TYPE ',I3,
     2' NUMBER ',I4,/,'  WITH A VALUE OF',F14.7)
      DO 171 I= 1,IHP1
171      V1(I)= X(I)- ACC(3,I)
      DO 175 I= 1,4
175      DOT(I)= 0.0
      IF (MPD.EQ.-1) GO TO 180
      DO 172 I= 1,KNU
         IK= I+ KMU
         DOT(1)= DOT(1)+ G1(JJ,I)*ACC(3,IK)
         DOT(2)= DOT(2)+ G1(JJ,I)*V1(IK)
172   CONTINUE
      STR= (-BA(JJ)-DOT(1))/DOT(2)
      WRITE (6,185) G1(JJ,1),BA(JJ),DOT(1),DOT(2),STR
      DO 173 I= 1,IHP1
173      X(I)= ACC(3,I)+ STR*V1(I)
      GO TO 264
180   DO 183 I= 1,KMU
         DOT(1)= DOT(1)+ G2(JJ,I)*ACC(3,I)
         DOT(2)= DOT(2)+ G2(JJ,I)*V1(I)
183   CONTINUE
      STR= (-BB(JJ)-DOT(1))/DOT(2)
      WRITE (6,185) G2(JJ,1),BB(JJ),DOT(1),DOT(2),STR
185   FORMAT(1X,' G2, BB, DOT1, DOT2, STR :',5F13.6)
```

```
      DO 186 I= 1,IHP1
186        X(I)= ACC(3,I)+ STR*V1(I)
      GO TO 264
  300 WRITE (6,980)
  850 FORMAT(1X,'THE CURRENT QUADRATIC APPRUXIMATION TO THE CURVE IS:'
     1,/,1X,'       F(X)                               A2(*U)+              A3')
  855 FORMAT (1X,3F17.7)
  935 FORMAT(1X,'ITERATION ',I4,' X= ', 9F10.5)
  936 FORMAT(1X,'NORM(G(X))= ',F11.6,'  G= ',8F11.6)
  940 FORMAT  (1X,//'****ALGORITHM BONBED WITH SINGULAR JACOBIAN****')
  938 FORMAT  (1X,' ITERATION : ',I4,' X= ',9F10.5)
  939 FORMAT  (1X,'NORM(F(X))= ',F11.6,' F= ',9F10.6)
  945 FORMAT(1X,'NO DESCENT POSSIBLE -- ALPHA WAS ZERO.')
  955 FORMAT(1X,' STEP SIZE= ALPHA= ',F9.7)
  980 FORMAT(1X,'NEWTON"S METHOD FAILED TO CONVERGE.')
      STOP
      END
C      QUADS FINDS THE SMALLEST NONNEGATIVE ROOT OF
C     U(1)*ALPHA**2 + U(2)*ALPHA + U(3) = 0, IF THERE
C     IS ONE.  IF NOT, IMAG IS SET = 1.
C
      SUBROUTINE QUADS(U,IMAG,ALPHA,BETA)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      COMMON/TOLER/ TOLFZ,TOLBC,TOLCV,THETA,STPMX,STPRD
      COMMON/DIM/ IH,N,M,KMJ,KNU,MP1,NM,ITAIL
      DIMENSION U(3)
      IMAG= 0
      IF (DABS(U(1)).GT. TOLFZ) GO TO 10
      IF (DABS(U(2)).LT.TOLFZ) GO TO 8
      ALPHA = -U(3)/U(2)
      BETA= ALPHA
      RETURN
    8 IMAG= 1
      RETURN
   10 RT= U(2)**2 - 4*U(1)*U(3)
      IF (RT) 20, 30, 40
C
C      THE QUADRATIC DOES NOT INTERSECT THIS CONSTRAINT.
C
   20 IMAG= 1
      RETURN
C
C     THERE IS A DOUBLE ROOT.
C
   30 ALPHA = U(2)/(U(1)*2.)
      BETA= ALPHA
      RETURN
C
C      TWO REAL ROOTS EXIST.
C
   40 DIS= DSQRT(RT)
      ALPHA= (-U(2)-DIS)/(2.*U(1))
      BETA= (-U(2)+DIS)/(2.*U(1))
      IF (ALPHA .LE. BETA) RETURN
      SAVE= ALPHA
      ALPHA = BETA
      BETA= SAVE
      RETURN
      END
C
      SUBROUTINE DSENT(ALPHA.PNORM)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/NEWT/ H(10.11).X(10).Z(10).ACC(3.10).BLAM(10)
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      DIMENSION Y(10),F(10)
    1 DO 5 I= 1,IH
    5 Y(I)= X(I)- ALPHA*Z(I)
      CALL FTN(F,Y)
      CALL NORM (F,S2,IH)
      IF (S2 .LT. PNORM) RETURN
      ALPHA= ALPHA/2.
      IF (ALPHA .GT. .00001) GO TO 1
      RETURN
      END
C
      SUBROUTINE GTN(IFL,Q,U,F)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      DIMENSION U(10),Q(10),F(10)
      Q(IHP1)= 0.0
      CALL FTN(F,U)
      THETA= U(IHP1)
      IF (KNU.EQ.0) GO TO 21
      DO 20 I= 1,KNU
          IK= I+ KMU
          QS= F(IK)+ U(IK)
          Q(IK)= THETA * QS - U(IK)
   20 CONTINUE
   21 IF (KMU.EQ.0) GO TO 24
      DO 23 I= 1,KMU
          IP= MU(I)
          IDB= KINBAS(IP)
          SUM= 0.0
          IF (KNU .EQ. 0) GO TO 7
          DO 3 K= 1,KNU
              IK= K+ KMU
    3         SUM= SUM+ G1(IDB,K) * U(IK)
    7     BLAM(I)= SUM + BA(IDB)
          DIF= BLAM(I)
          QS= F(I)+ DIF
          Q(I)= THETA*QS - DIF
   23 CONTINUE
   24 IF (IFL.EQ. 0) RETURN
      KFUN= KFUN+ 1
      SUM= 0.0
      IF (IFL.EQ.2) GO TO 31
      DO 25 I= 1,IHP1
          SUM= SUM + ACC(2,I) * U(I)
   25 CONTINUE
      Q(IHP1)= SUM - BT
      RETURN
   30 IF (MPD.EQ.-1) GO TO 50
      IF (KNU.EQ.0) GO TO 45
      DO 40 I= 1,KNU
         IK= I+ KMU
         SUM= SUM+ G1(JJ,I)*U(IK)
   40 CONTINUE
   45 Q(IHP1)= SUM+ BA(JJ)
      RETURN
   50 IF(KMU.EQ.0) GO TO 60
      DO 55 I= 1,KMU
         SUM= SUM+ G2(JJ,I)*U(I)
   55 CONTINUE
   60 Q(IHP1)= SUM+ BB(JJ)
      RETURN
      END
C
C  SUBROUTINE DERIVG CALCULATES THE HESSIAN OF THE
C  RETRACTION FUNCTION, PLUS SOME OTHER GRADIENT
C  SPECIFIED BY IFL.
C      IFL=0 :   NO OTHER GRADIENT
C      IFL=1 :   A SPHERE DETERMINES THE GRADIENT.
C      IFL=2 :   ONE OF THE LINEAR CONSTRAINTS IS
C                THE GRADIENT.
C
      SUBROUTINE DERIVG(IFL,U,G,F)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),SIGMA(952),KINBAS(1302),IDBAS(1302)
      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX2/ JH,SIGMA,KINBAS,IDBAS
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      DIMENSION G(10),U(10),Q(10),F(10)
      THETA= U(IHP1)
      TMTH= 1. - THETA
      CALL DERIV
      DO 20 J= 1,IH
         DO 10 I= 1,IH
            H(I,J)= THETA * H(I,J)
   10    CONTINUE
   20 CONTINUE
      DO 26 I= 1,IH
         IF(I .GT. KMU) GO TO 25
         IP= MU(I)
         IDB= KINBAS(IP)
         DO 24 J= 1,KNU
            JK= J+ KMU
            H(I,JK)= H(I,JK) - TMTH*G1(IDB,J)
   24    CONTINUE
         GO TO 28
   25    H(I,I)= H(I,I)- TMTH
   26 CONTINUE
      DO 35 I= 1,IH
         IF(I .GT. KMU) GO TO 30
         H(I,IHP1)= F(I)+ BLAM(I)
         GO TO 35
   30    H(I,IHP1)= F(I) + U(I)
   35 CONTINUE
   39 IF (IFL .EQ. 0) RETURN
      IF (IFL.EQ.2) GO TO 50
      DO 40 I= 1,IHP1
   40       H(IHP1,I)=  ACC(2,I)
      RETURN
   50 IF (MPD.EQ.-1) GO TO 70
      DO 60 J= 1,IHP1
            H(IHP1,J)= 0.0
            IF (J.LE.KMU) GO TO 60
            IF (J .EQ. IHP1) GO TO 60
            JM= J- KMU
            H(IHP1,J)= G1(JJ,JM)
   60 CONTINUE
      RETURN
   70 DO 80 J= 1,IHP1
            H(IHP1,J)= 0.0
            IF (J.GT.KMU) GO TO 80
            H(IHP1,J)= G2(JJ,J)
   80 CONTINUE
      RETURN
      END
      SUBROUTINE CONCHK(GMIN,KGMIN,MP2)
C
C        THIS SUBROUTINE EVALUATES THE CONSTRAINT FUNCTIONS
C  AND FINDS THE SMALLEST VALUE IN GMIN.  ZFLAG IS BOTH AN
C  INPUT AND OUTPUT PARAMATER.  IF ZFLAG.EQ.1 INITIALLY, THEN
C  ONLY THE NONLINEAR CONSTRAINT IS EVALLATED.  ZFLAG IS SET
C  EQUAL TO 2 IF ANY CONSTRAINT IS NONPOSITIVE, OTHERWISE IT
C  REMAINS ZERO.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 C(800)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LS,IA,IE,PUN,LC(20),IC(800)
      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/LP1/PI(1302),XX(1302)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
      COMMON/DIM/ IH,N,M,KMU,KNU,NP1,NM,ITAIL
      GMIN= 10.
      DO 20 I= 1,M
         IK= JH(I)
         GG= 0.0
         IF (KNU.EQ.0) GO TO 15
         DO 10 J= 1,KNU
            JK= J+ KMU
            GG= GG+ G1(I,J)*X(JK)
   10    CONTINUE
   15    GG= GG+ BA(I)
         XX(IK)= GG
         IF (IK.EQ.M .OR. IK.LE.IH) GO TO 20
         IF (MFLAG.EQ.3 .AND. IK.EQ.NM) GO TO 20
         IF (GG.GE.GMIN) GO TO 20
         GMIN= GG
         KGMIN= I
         MP2= 1
   20 CONTINUE
      DO 40 I= 1,N
         IK= DIGMA(I)
         GG= 0.0
         IF (KMU.EQ.0) GO TO 35
         DO 30 J= 1,KMU
            GG= GG+ G2(I,J)*X(J)
   30     CONTINUE
   35    GG= GG+ BB(I)
         PI(IK)= GG
         IF (IK .EQ. M) GO TO 40
         IF (GG.GE.GMIN) GO TO 40
         GMIN= GG
         KGMIN= I
         MP2= -1
   40 CONTINUE
      RETURN
      END
C
C    SUBROUTINE FTN EVALUATES THE FUNCTION F(X)=
C     <B(MU)+ D1*X(MU) + DIAG(X(MU))*(F1*X(NU))        >
C     <B(NU)+ D2*X(MU) + DIAG(X(NU))*(F2*X(MU)+ F3)   >
C
      SUBROUTINE FTN(F,Y)
C
C       THIS ROUTINE EVALUATES THE FUNCTION WHICH THE ENDPOINT
C  SUBROUTINE IS CURRENTLY TRYING TO FIND A ROOT OF.  IF MFLAG
C  EQUALS 1 THEN THE LAST FUNCTIONAL COMPRISING F IS ONE OF THE
C  INEQUALITY CONSTRAINTS-- THE INEQUALITY DETERMINED BY THE
C  PARAMETERS MPD AND JJ.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 C(800)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/INDX1/ NUH(10),MUF(10),NU(10),MU(10)
      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      DIMENSION F(IH),Y(IH)
      KMU1= KMU+ 1
      KFUN= KFUN+ IH
      IF (KMU.EQ.0) GO TO 23
      DO 20 I= 1,KMU
      DOT= 0.0
      IF (KNU.EQ.0) GO TO 15
      DO 10 J= 1,KNU
      JK= J+ KMU
   10 DOT= DOT+ F1(I,J)*Y(JK)
   15 F(I)= -Y(I)*(DOT+ E1(I))+ BF1(I)
      IF(KMU.EQ.0) GO TO 20
      DO 18 J= 1,KMU
   18 F(I)= F(I)+ D1(I,J)*Y(J)
   20 CONTINUE
   23 IF (KNU.EQ.0) RETURN
      DO 40 I= KMU1,IH
      IMB= I- KMU
      DOT= 0.0
      IF(KMU.EQ.0) GO TO 30
      DO 25 J= 1,KMU
   25 DOT= DOT+ F2(IMB,J)*Y(J)
   30 F(I)= -Y(I)*(DOT+E2(IMB))+ BF2(IMB)
      IF (KMU.EQ.0) GO TO 40
      DO 32 J= 1,KMU
   32 F(I)= F(I)+ D2(IMB,J)*Y(J)
   40 CONTINUE
      RETURN
      END
      SUBROUTINE DERIV
C
C    THIS SUBROUTINE CALCULATES THE EXACT JACOBIAN OF THE
C    BILINEAR FUNCTION DEFINED BY BLCONS.  THERE MAY OR MAY
C    NOT BE ANOTHER FUNCTIONAL APPENDED WHICH WE ARE ATTEMPT-
C    ING TO MAKE BINDING WITH NEWTON'S METHOD.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 C(800)
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),SIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(800)
      COMMON/NEWT/ H(10,11),X(10),Z(10),ACC(3,10),BLAM(10)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      KJAC= KJAC+ 1
      DO 10 I= 1,IH
          DO 10 J= 1,IH
   10         H(I,J)= 0.
      KMU1= KMU+1
      KNU1= KNU+ 1
      IH1= IH- 1
      IF (KMU.EQ.0) GO TO 55
      DO 50 J= 1,KMU
          DO 30 I= 1,KMU
              H(I,J)= D1(I,J)
              IF (I.NE.J) GO TO 30
              DOT= 0.0
              IF (KNU.EQ.0) GO TO 25
              DO 20 K= 1,KNU
              KB= KMU+K
   20             DOT= DOT+ F1(I,K)*X(KB)
   25         H(I,J)= H(I,J)- DOT- E1(I)
   30     CONTINUE
          IF (KNU.EQ.0) GO TO 50
          DO 40 I= KMU1,IH
          IMB= I- KMU
              H(I,J)= D2(IMB,J)- F2(IMB,J)*X(I)
   40     CONTINUE
   50 CONTINUE
   55 IF (KMU.EQ.0) RETURN
      DO 70 J= KMU1,IH
          JMU= J-KMU
      IF (KMU.EQ.0) GO TO 72
          DO 60 I= 1,KMU
```

```
              H(I,J)= - X(I)*F1(I,JMU)
   60     CONTINUE
   70 CONTINUE
   72 IF (KNU.EQ.0) RETURN
      DO 80 I= 1,KNU
      IK= KMU+I
      IF (KMU.EQ.0) GO TO 77
      DO 75 K= 1,KMU
   75 H(IK,IK)= H(IK,IK)- F2(I,K)*X(K)
   77 H(IK,IK)= H(IK,IK)- E2(I)
   80 CONTINUE
      RETURN
      END
      SUBROUTINE NORM(Y,S1,N)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(10)
      S1= 0.0
      DO 10 I= 1,N
   10     S1= S1+ Y(I)*Y(I)
      S1= DSQRT(S1)
      RETURN
      END
      SUBROUTINE DECOMP(NN,A,UL)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/INT/ IPS(30),KDET,KCUNT,ISING,KEND
      DIMENSION A(10,10),UL(10,10),SCALES(10)
      N= NN
      KDET= 1
C
C   INITIALIZE IPS,UL, AND SCALES.
      ISING= 0
      DO 5 I= 1,N
          IPS(I)= I
          ROWNRM= 0.0
          DO 2 J= 1,N
              UL(I,J)= A(I,J)
              IF (ROWNRM-DABS(UL(I,J))) 1,2,2
    1         ROWNRM= DAPS(UL(I,J))
    2     CONTINUE
          IF (ROWNRM) 3,4,3
    3     SCALES(I)= 1./ROWNRM
          GO TO 5
    4     CALL SING(1)
          ISING= 1
          SCALES(I)= 0.
    5 CONTINUE
C
C   GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
      NM1= N- 1
      DO 17 K= 1,NM1
          BIG= 0.0
          DO 11 I= K,N
              IP= IPS(I)
              SIZE= DABS(UL(IP,K))*SCALES(IP)
              IF (SIZE-BIG)  11,11,10
   10                BIG= SIZE
                     IDXPIV= I
   11     CONTINUE
          IF (BIG)  13,12,13
   12         CALL SING(2)
```

```
              ISING= 1
              GO TO 17
    13    IF (IDXPIV-K) 14,15,14
    14        J= IPS(K)
              IPS(K)= IPS(IDXPIV)
              IPS(IDXPIV)= J
              KDET= -KDET
    15    KP= IPS(K)
          PIVOT= UL(KP,K)
          KP1= K+1
          DO 16 I= KP1,N
              IP= IPS(I)
              EM= -UL(IP,K)/PIVOT
              UL(IP,K)= -EM
              DO 16 J= KP1,N
                  UL(IP,J)= UL(IP,J)+ EM*UL(KP,J)
    16        CONTINUE
    17 CONTINUE
      KP= IPS(N)
      IF (UL(KP,N)) 19,18,19
    18 CALL SING(2)
      ISING= 1
    19 RETURN
      END
C
C
      SUBROUTINE SOLVE(NN,UL,B,X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/INT/ IPS(30),KDET,KCUNT,ISING,KEND
      DIMENSION UL(10,10),B(10),X(10)
      N= NN
      NP1= N+ 1
      IP= IPS(1)
      X(1)= B(IP)
      DO 2  I= 2,N
          IP= IPS(I)
          IM1= I-1
          SUM= 0.0
          DO 1 J= 1,IM1
    1         SUM= SUM+ UL(IP,J)*X(J)
    2 X(I)= B(IP)- SUM
C
      IP= IPS(N)
      X(N)= X(N)/UL(IP,N)
      IF(UL(IP,N).GE.0) GO TO 10
          KDET= -KDET
   10 DO 4 IBACK= 2,N
      I= NP1- IBACK
C     I GOES (N-1),....1
          IP= IPS(I)
          IP1= I+1
          SUM= 0.0
          DO 3 J= IP1,N
    3         SUM= SUM+ UL(IP,J)*X(J)
          DIV= UL(IP,I)
          IF(DIV.GE.0) GO TO 4
          KDET= -KDET
    4 X(I)= (X(I)-SUM)/DIV
      RETURN
      END
```

```
C
C
      SUBROUTINE SING(IWHY)
   11 FORMAT(1X,'MATRIX WITH ZERO ROW IN DECOMPOSE.')
   12 FORMAT(1X,'SINGULAR MATRIX IN DECOMPOSE. ZERO DIVIDE IN SOLVE')
      IF (IWHY- 1) 1,1,2
    1 WRITE (6,11)
      GO TO 10
    2 WRITE (6,12)
   10 RETURN
      END
      SUBROUTINE DEBUG(MODE)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MIN
      REAL*4 TIME
      INTEGER PD,PD1,PD2,S,R,SS,RR,ZFLAG,RS,P
      INTEGER*2 JH(350),DIGMA(952),KINBAS(1302),IDBAS(1302)
      INTEGER*2 ISTYPE,LA,LE,IA,IE,PUN,LC(20),IC(80C)
      DOUBLE PRECISION  E(8000)
      REAL  A(4000),C(800),CMIN,COND,ERMAX,SUMINF
C
      COMMON DSUM,DPROD,DY,DE,DP,B(350),X(350),Y(350),YTEMP(350),
     1A,E,CMIN,COND,ERMAX,SUMINF,ICNAM(1302,2),NAME(20),
     2NTEMP(20),KINP,ITIM,JTIM,ITINV,JTINV,MSTAT,IOBJ,IROWP,IVIN,IVOUT,
     3ITCNT,INVFRQ,ITRLIM,IFFEZ,JCOLP,NROW,NCOL,NELEM,NETA,NLELEM,NLETA,
     4NGELEM,NINF,NUELEM,NUETA,NNEGDJ,NLINES,ISTYPE(35C),
     5LA(1302),LE(2002),PUN(8),
     6IPUNC,NDEGI,NDUAL,NIPIW,IFBAS,IFCRSH
      COMMON ITCH,ITCHA,IFPIWT,IFNEG,KOUTB
      COMMON IA(4000),IE(8000)
      COMMON/LPI/PI(1302),XX(1302)
      COMMON/BLCST/BF1(10),BF2(10),E1(10),E2(10),C,IC,LC
      COMMON/BLCST2/D1(10,10),D2(9,10),F1(9,10),F2(9,10)
      COMMON/LNCONS/G1(350,10),G2(400,10),BA(350),BB(400)
      COMMON/INDX1/ NUH(10),MUH(10),NU(10),MU(10)
      COMMON/INDX2/ JH,DIGMA,KINBAS,IDBAS
      COMMON/SCAL/ BT,NB,JJ,MFLAG,IHP1,P,PD,MPD,KFUN,KJAC
      COMMON/DIM/ IH,N,M,KMU,KNU,MP1,NM,ITAIL
      IF (MODE-2) 10,30,60
   10 TIME= 1.0
      WRITE(6,20) TIME
   20 FORMAT (1X,' THE TIME LEFT IS NOW ',F9.5,' SEC.')
      RETURN
   30 WRITE (6,40)
      WRITE (6,41)  (BA(I),I= 1,M)
      WRITE (6,49) (BB(I),I=1,N)
      IF (KNU.EQ.0) GO TO 35
      WRITE (6,42) ((G1(I,J),I= 1,M),J= 1,KNU)
   35 IF (KMU.EQ.0) RETURN
      WRITE (6,43) ((G2(I,J),I= 1,N),J= 1,KMU)
      RETURN
   40 FORMAT (/,1X,' LINEAR CONSTRAINTS.')
   42 FORMAT (1X,' G1= ',7F13.5)
   41 FORMAT (1X,' BA= ',7F13.5)
   49 FORMAT (1X,' BB= ',7F13.5)
   43 FORMAT (1X,' G2= ',7F13.5)
   60 IF (MODE - 4) 65,70,100
   65 WRITE (6,80)  (KINBAS(I),I=1,NCOL)
      WRITE (6,85) (JH(I),I= 1,NROW)
   80 FORMAT (1X,' KINBAS= ',20I4)
   85 FORMAT (1X.' JH= '.20I4)
      RETURN
   70 WRITE (6.71) KNU.KMU
      WRITE (6.152) (NUH(I).I=1.IH)
      WRITE (6.151) (MUH(I).I=1.IH)
      WRITE (6.74) (XX(I).I=1.IH)
      WRITE (6,75) (PI(I).I=1.IH)
   71 FORMAT (//.' SUPERBASIC INFO: KNU='.I3.'  KMU= '.I3)
   74 FORMAT ('    T     ='.9F12.6)
   75 FORMAT (' LAMBDA='.9F12.6)
      RETURN
  100 IF (MODE - 6)  101.140.140
  101 WRITE (6.120)  KNU,KMU
      IF (KMU.EQ.0) GO TO 105
      WRITE (6.121) (BF1(I).I=1.KMU)
      WRITE (6.122) ((D1(I.J).J=1.KMU).I=1.KMU)
      WRITE (6,123) (E1(I).I=1.KMU)
      IF (KNU.EQ.0) GO TO 105
      WRITE (6,124) ((F1(I.J).J=1.KNU).I=1.KMU)
  105 IF (KNU.EQ.0) RETURN
      WRITE (6.125) (BF2(I).I=1.KNU)
      WRITE (6.126) (E2(I).I=1.KNU)
      IF (KMU.EQ.0) RETURN
      WRITE (6.127) ((D2(I.J).J=1.KMU).I=1.KNU)
      WRITE (6.128) ((F2(I.J).J=1.KMU).I=1.KNU)
      RETURN
  120 FORMAT (1X.'BL CONSTRAINTS.  KNU.KMU:'.2I3)
  121 FORMAT (1X.' BF1= '. F14.6)
  122 FORMAT (1X.' D1= '.F14.6)
  123 FORMAT (1X.' E1= '.F14.6)
  124 FORMAT (1X.' F1= '.F14.6)
  125 FORMAT (1X.' BF2='.F14.6)
  126 FORMAT (1X.' E2= '.F14.6)
  127 FORMAT (1X.' D2= '.F14.6)
  128 FORMAT (1X.' F2= '.F14.6)
  140 WRITE (6.150) PO.KMU.KNU.IHC
      WRITE (6.151) (MUH(I).I= 1.IH)
      WRITE (6.152) (NUH(I).I= 1.IH)
  150 FORMAT (//.1X.' ENDPOINT FINDER DATA:PO.KMU.KNU.IHC= '.5I3)
  151 FORMAT (1X.'MUH(I)= '.20I3)
  152 FORMAT (1X.'NUH(I)= '.20I3)
      WRITE (6.154) (JH(I).I=1.M)
      WRITE (6.155) (DIGMA(I).I=1.N)
  154 FORMAT (1X.'JH= '.15I4)
  155 FORMAT (1X.'DIGMA= '.15I4)
      WRITE (6.153) (XX(I).I=1.NM)
      WRITE (6.156) (PI(I).I=1,NM)
  153 FORMAT (1X.' XX= '.9F13.6)
  156 FORMAT (1X.' PI= '.9F13.6)
      RETURN
      END
```

References

[1] Elken, T.R., "On The Solution of Nonlinear Equations by Path Methods," TR SOL 77-25, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, October 1977.

[2] Elken, T.R., "The Computation of Economic Equilibria by Path Methods," TR SOL 77-26, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, October 1977.

[3] Forsythe, G.E., The Solution of Linear Systems of Equations, New York: Prentice Hall.

[4] Tomlin, J.A., "Users Guide for LPM1," unpublished report, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, 1975.

[5] Tomlin, J.A., "Users Guide to LCPL -- A Program for Solving Linear Complementarity Problems by Lemke's Method," TR SOL 76-16, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, August 1976.

[6] Tomlin, J.A., "Programmer's Guide to LCPL -- A Program for Solving Linear Complementarity Problem's by Lemke's Method," TR SOL 76-25, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, October 1976.

[7] Wilson, R., "The Bilinear Complementarity Problem and Competitive Equilibria of Linear Economic Models," TR SOL 76-2, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, January 1976.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
SOL-78-17 ✓

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
6
BCA and HRA: Two Programs for Computing Economic Equilibria.

5. TYPE OF REPORT & PERIOD COVERED
9
Technical Report

6. PERFORMING ORG. REPORT NUMBER
SOL 78-17

7. AUTHOR(s)
10
Thomas Elken

8. CONTRACT OR GRANT NUMBER(s)
15
NØØ014-75-C-Ø267,
EY-76-S-Ø3-Ø326

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Operations Research -- SOL
Stanford University
Stanford, CA 94305

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NR-047-143

11. CONTROLLING OFFICE NAME AND ADDRESS
Operations Research Program -- ONR
Department of the Navy, 800 N. Quincy Street
Arlington, VA 22217

12. REPORT DATE
11
August 1978

13. NUMBER OF PAGES
128

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
12 132p.

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Economic Equilibria
Linear Programming
Bilinear Complementarity
Homotopy Retraction Algorithm
Path-Following Methods

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report gives programmers information useful for the utilization and modification of BCA and HRA, two programs for computing economic equilibria. The BCA implements the bilinear complementarity algorithm and HRA a homotopy retraction algorithm. The report describes the subroutines and variables used as well as the usual documentation: input format, sample input and output for two problems, and program listings.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 765

46